QuickTime Music Architecture Reference

QuickTime > Audio

2006-05-23

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

QuickTime Music Architecture Reference

Framework:	Frameworks/QuickTime.framework
Declared in	IOMacOSTypes.h QuickTimeMusic.h

Overview

The QuickTime Music Architecture (QTMA) allows QuickTime movies, applications, and other software to play individual musical notes, sequences of notes, and a broad range of sounds from a variety of instruments and synthesizers. With QTMA, you can also import Standard MIDI files and convert them into a QuickTime movie for easy playback.

Functions by Task

Allocating and Using Note Channels

`NADisposeNoteChannel` (page 86) Deprecated in Mac OS X v10.5
Deletes a specified note channel.

`NAFindNoteChannelTone` (page 86) Deprecated in Mac OS X v10.5
Locates the instrument that best fits a requested tone description for a specific channel.

`NAGetController` (page 87) Deprecated in Mac OS X v10.5
Retrieves the controller settings for a note channel.

`NAGetIndNoteChannel` (page 88) Deprecated in Mac OS X v10.5
Returns the number of note channels handled by the specified note allocator instance.

`NAGetKnob` (page 88) Deprecated in Mac OS X v10.5
Obtains the value of a knob for a given note channel.

`NAGetNoteChannelInfo` (page 90) Deprecated in Mac OS X v10.5
Returns the index of the music component for the allocated channel and its part number on that music component.

`NAGetNoteRequest` (page 91) Deprecated in Mac OS X v10.5
Retrieves the NoteRequest structure that was passed to a note channel.

`NANewNoteChannel` (page 93) Deprecated in Mac OS X v10.5
Requests a new note channel with the qualities described in a NoteRequest structure.

`NANewNoteChannelFromAtomicInstrument` (page 93) Deprecated in Mac OS X v10.5
Requests a new note channel for an atomic instrument.

NAPlayNote (page 98) Deprecated in Mac OS X v10.5
Plays a note with a specified pitch and velocity on the specified note channel.

NAPrerollNoteChannel (page 99) Deprecated in Mac OS X v10.5
Attempts to reallocate the note channel if it was invalid previously.

NAResetNoteChannel (page 100) Deprecated in Mac OS X v10.5
Turns off all currently active notes on the note channel and resets all controllers to their default values.

NASendMIDI (page 101) Deprecated in Mac OS X v10.5
Sends a MIDI music packet to a synthesizer that contains a specific note channel.

NASetAtomicInstrument (page 102) Deprecated in Mac OS X v10.5
Initializes a synthesizer part with an atomic instrument.

NASetController (page 103) Deprecated in Mac OS X v10.5
Changes the controller setting on a note channel to a specified value.

NASetInstrumentNumber (page 104) Deprecated in Mac OS X v10.5
Initializes initializes a synthesizer part with the specified instrument.

NASetInstrumentNumberInterruptSafe (page 104) Deprecated in Mac OS X v10.5
Initializes a synthesizer part with the specified instrument during interrupt time.

NASetKnob (page 105) Deprecated in Mac OS X v10.5
Sets a note channel knob to a particular value.

NASetNoteChannelBalance (page 106) Deprecated in Mac OS X v10.5
Modifies the pan controller setting for a note channel.

NASetNoteChannelSoundLocalization (page 107) Deprecated in Mac OS X v10.5
Passes sound localization data to a note channel.

NASetNoteChannelVolume (page 107) Deprecated in Mac OS X v10.5
Sets the volume on the specified note channel.

NAUnrollNoteChannel (page 110) Deprecated in Mac OS X v10.5
Marks a note channel as available to be stolen.

Calling Generic Music Component Clients

MusicDerivedSetInstrument (page 48) Deprecated in Mac OS X v10.5
The complete instrument defined by the Part structure to the synthesizer.

MusicDerivedSetKnob (page 49) Deprecated in Mac OS X v10.5
Called when any of the synthesizer's knobs are altered.

MusicDerivedSetMIDI (page 50) Deprecated in Mac OS X v10.5
Sets the MIDI channel and other MIDI settings for MIDI output only.

MusicDerivedSetPart (page 51) Deprecated in Mac OS X v10.5
Sets the polyphony for the part specified in the GCPart structure.

Managing Instruments and Parts

MusicGetInstrumentAboutInfo (page 60) Deprecated in Mac OS X v10.5
Obtains the information about an instrument that appears in its About box.

MusicGetInstrumentInfo (page 61) Deprecated in Mac OS X v10.5
Obtains a list of instruments supported by a synthesizer.

MusicGetPart (page 67) Deprecated in Mac OS X v10.5
Returns the MIDI channel and maximum polyphony for a particular part.

MusicGetPartAtomicInstrument (page 68) Deprecated in Mac OS X v10.5
Returns the atomic instrument currently in a part.

MusicGetPartController (page 69) Deprecated in Mac OS X v10.5
Returns the value of a specified controller on a specified part.

MusicGetPartInstrumentNumber (page 70) Deprecated in Mac OS X v10.5
Returns the instrument number currently assigned to a part.

MusicGetPartKnob (page 70) Deprecated in Mac OS X v10.5
Retrieves the current value of a knob for a part.

MusicGetPartName (page 71) Deprecated in Mac OS X v10.5
Returns the string name of a part.

MusicResetPart (page 72) Deprecated in Mac OS X v10.5
Silences all sounds on a specified part and resets all controllers on that part to their default values.

MusicSetPart (page 76) Deprecated in Mac OS X v10.5
Sets the MIDI channel and maximum polyphony for a specified part.

MusicSetPartAtomicInstrument (page 77) Deprecated in Mac OS X v10.5
Initializes a part with an atomic instrument.

MusicSetPartController (page 78) Deprecated in Mac OS X v10.5
Initializes the value of a specified controller on a specified part.

MusicSetPartInstrumentNumber (page 79) Deprecated in Mac OS X v10.5
Superseded by MusicSetPartInstrumentNumberInterruptSafe.

MusicSetPartInstrumentNumberInterruptSafe (page 79) Deprecated in Mac OS X v10.5
Initializes a part with a particular instrument.

MusicSetPartKnob (page 80) Deprecated in Mac OS X v10.5
Sets a knob for a specified part.

MusicSetPartName (page 80) Deprecated in Mac OS X v10.5
Changes the name of an instrument in a specified part.

MusicSetPartSoundLocalization (page 81) Deprecated in Mac OS X v10.5
Passes sound localization data to a specified synthesizer part.

MusicStorePartInstrument (page 83) Deprecated in Mac OS X v10.5
Puts whatever instrument is on the specified part into the synthesizer's instrument store.

Managing Synthesizers

MusicFindTone (page 53) Deprecated in Mac OS X v10.5
Returns the number of the best-matching instrument provided by a specified music component.

MusicGetDescription (page 57) Deprecated in Mac OS X v10.5
Returns a structure describing the synthesizer controlled by the music component device.

MusicGetDeviceConnection (page 58) Deprecated in Mac OS X v10.5
Determines how many hardware synthesizers are available to a music component and gets the IDs for those devices.

`MusicGetDrumKnobDescription` (page 58) Deprecated in Mac OS X v10.5
Returns a description of a drum kit knob.

`MusicGetInstrumentKnobDescription` (page 62) Deprecated in Mac OS X v10.5
Obtains the description of an instrument knob.

`MusicGetKnob` (page 63) Deprecated in Mac OS X v10.5
Returns the value of the specified global synthesizer knob.

`MusicGetKnobDescription` (page 64) Deprecated in Mac OS X v10.5
Returns a pointer to an initialized knob description structure describing a global synthesizer knob.

`MusicGetKnobSettingStrings` (page 65) Deprecated in Mac OS X v10.5
Returns a list of knob setting names known by the specified music component.

`MusicGetMIDIPorts` (page 66) Deprecated in Mac OS X v10.5
Returns the number of input and output ports a MIDI device has.

`MusicGetMIDIProc` (page 67) Deprecated in Mac OS X v10.5
Returns a pointer to the procedure a music component is using to process external MIDI notes.

`MusicPlayNote` (page 71) Deprecated in Mac OS X v10.5
Plays a note on a specified part at a specified pitch and velocity.

`MusicSendMIDI` (page 73) Deprecated in Mac OS X v10.5
Sends a MIDI packet to a specified port.

`MusicSetKnob` (page 74) Deprecated in Mac OS X v10.5
Modifies the value of the specified global synthesizer knob.

`MusicSetMIDIProc` (page 75) Deprecated in Mac OS X v10.5
Informs the music component what procedure to call when it needs to send MIDI data.

`MusicUseDeviceConnection` (page 84) Deprecated in Mac OS X v10.5
Tells a music component which hardware synthesizer to talk to.

Managing the Generic Music Component

`MusicGenericConfigure` (page 54) Deprecated in Mac OS X v10.5
Informs the generic music component what services your music component requires and points to any resources that are necessary.

MIDI Component Functions

`QTMIDIGetMIDIPorts` (page 112) Deprecated in Mac OS X v10.5
Returns two lists of MIDI ports supported by the specified MIDI component: a list of ports that can receive MIDI input and a list of ports that can send MIDI output.

`QTMIDISendMIDI` (page 113) Deprecated in Mac OS X v10.5
Sends MIDI data to a MIDI port.

`QTMIDIUseSendPort` (page 114) Deprecated in Mac OS X v10.5
Allocates a MIDI port for output or to release the port.

Miscellaneous Music Component Functions

`MusicGetMasterTune` (page 65) Deprecated in Mac OS X v10.5
Returns the synthesizer's master tuning as a fixed-point value in semitones.

`MusicSetMasterTune` (page 74) Deprecated in Mac OS X v10.5
Alters a synthesizer's master tuning.

`MusicSetOfflineTimeTo` (page 76) Deprecated in Mac OS X v10.5
Advances the synthesizer clock when the synthesizer is not running in real time.

`MusicStartOffline` (page 82) Deprecated in Mac OS X v10.5
Informs the QuickTime music synthesizer that the music will not be played through the speakers.

`MusicTask` (page 83) Deprecated in Mac OS X v10.5
Allows a music component to perform tasks it must perform at foreground task time.

Note Allocator Configuration and Utilities

`NAGetMIDIPorts` (page 89) Deprecated in Mac OS X v10.5
The MIDI input and output ports available to a note allocator.

`NAGetRegisteredMusicDevice` (page 91) Deprecated in Mac OS X v10.5
Returns details about music components registered to the specified note allocator instance.

`NARegisterMusicDevice` (page 99) Deprecated in Mac OS X v10.5
Registers a music component with the note allocator.

`NASaveMusicConfiguration` (page 101) Deprecated in Mac OS X v10.5
Saves the current list of registered devices to a file.

`NATask` (page 109) Deprecated in Mac OS X v10.5
Called periodically to allow the note allocator to perform tasks in foreground task time.

`NAUnregisterMusicDevice` (page 109) Deprecated in Mac OS X v10.5
Removes a previously registered music component from the note allocator.

Note Allocator Interface Tools

`NACopyrightDialog` (page 85) Deprecated in Mac OS X v10.5
Displays a copyright dialog box with information specific to a music device.

`NAPickArrangement` (page 94) Deprecated in Mac OS X v10.5
Displays a dialog box to allow instrument selection.

`NAPickEditInstrument` (page 95) Deprecated in Mac OS X v10.5
Presents a user interface for changing the instrument in a live note channel or modifying an atomic instrument.

`NAPickInstrument` (page 97) Deprecated in Mac OS X v10.5
Presents a user interface for picking an instrument.

`NAStuffToneDescription` (page 108) Deprecated in Mac OS X v10.5
Initializes a tone description structure with the details of a General MIDI note channel.

Using the Tune Player

`TuneGetIndexedNoteChannel` (page 115) Deprecated in Mac OS X v10.5
Determines how many parts a tune is playing and which instrument is assigned to those parts.

`TuneGetNoteAllocator` (page 115) Deprecated in Mac OS X v10.5
Returns the instance of the note allocator that the tune player is using.

`TuneGetPartMix` (page 116) Deprecated in Mac OS X v10.5
Gets volume, balance, and mixing settings for a specified part of a tune.

`TuneGetStatus` (page 117) Deprecated in Mac OS X v10.5
Returns an initialized structure describing the state of the tune player instance.

`TuneGetTimeBase` (page 117) Deprecated in Mac OS X v10.5
Returns the time base of the tune player.

`TuneGetTimeScale` (page 118) Deprecated in Mac OS X v10.5
Returns the current time scale for a specified tune player instance.

`TuneGetVolume` (page 118) Deprecated in Mac OS X v10.5
Returns the volume associated with an entire tune sequence.

`TuneInstant` (page 119) Deprecated in Mac OS X v10.5
Plays a particular sequence of events active at a specified position.

`TunePreroll` (page 120) Deprecated in Mac OS X v10.5
Prepares to play a tune player sequence data by attempting to reserve note channels for each part in the sequence.

`TuneQueue` (page 120) Deprecated in Mac OS X v10.5
Places a sequence of music events into a queue to be played.

`TuneSetBalance` (page 121) Deprecated in Mac OS X v10.5
Modifies the pan controller setting for a tune player.

`TuneSetHeader` (page 122) Deprecated in Mac OS X v10.5
Prepares the tune player to accept subsequent music event sequences by defining one or more parts to be used by sequence Note events.

`TuneSetHeaderWithSize` (page 123) Deprecated in Mac OS X v10.5
Similar to TuneSetHeader but lets you specify the header length.

`TuneSetNoteChannels` (page 124) Deprecated in Mac OS X v10.5
Assigns note channels to a tune player.

`TuneSetPartMix` (page 124) Deprecated in Mac OS X v10.5
Sets volume, balance, and mixing settings for a specified part of a tune.

`TuneSetPartTranspose` (page 125) Deprecated in Mac OS X v10.5
Modifies the pitch and volume of every note of a tune.

`TuneSetSofter` (page 126) Deprecated in Mac OS X v10.5
Adjusts the volume a tune is played at to the softer volume produced by QuickTime 2.1.

`TuneSetSoundLocalization` (page 127) Deprecated in Mac OS X v10.5
Passes sound localization data to a tune player.

`TuneSetTimeScale` (page 127) Deprecated in Mac OS X v10.5
Sets the time scale used by the specified tune player instance.

`TuneSetVolume` (page 128) Deprecated in Mac OS X v10.5
Sets the volume for an entire sequence.

TuneStop (page 128) Deprecated in Mac OS X v10.5
Stops a currently playing sequence.

TuneTask (page 129) Deprecated in Mac OS X v10.5
Lets a tune player to perform tasks it must perform at foreground task time.

TuneUnroll (page 129) Deprecated in Mac OS X v10.5
Releases any note channel resources that may have been locked down by previous calls to TunePreroll for this tune player.

Supporting Functions

DisposeMusicMIDISendUPP (page 45) Deprecated in Mac OS X v10.5
Disposes of a MusicMIDISendUPP pointer.

DisposeMusicOfflineDataUPP (page 45) Deprecated in Mac OS X v10.5
Disposes of a MusicOfflineDataUPP pointer.

DisposeTuneCallBackUPP (page 46) Deprecated in Mac OS X v10.5
Disposes of a TuneCallBackUPP pointer.

DisposeTunePlayCallBackUPP (page 46) Deprecated in Mac OS X v10.5
Disposes of a TunePlayCallBackUPP pointer.

MusicDerivedCloseResFile (page 47) Deprecated in Mac OS X v10.5
Closes a music movie resource file.

MusicDerivedMIDISend (page 47) Deprecated in Mac OS X v10.5
Sends a MIDI packet to a music component.

MusicDerivedOpenResFile (page 48) Deprecated in Mac OS X v10.5
Opens the music resource file for a music component.

MusicDerivedSetPartInstrumentNumber (page 51) Deprecated in Mac OS X v10.5
Sets the instrument specified in the GCPart structure.

MusicDerivedStorePartInstrument (page 52) Deprecated in Mac OS X v10.5
Undocumented

MusicGenericGetKnobList (page 55) Deprecated in Mac OS X v10.5
Gets a list of the knobs of a given type for the generic music component.

MusicGenericGetPart (page 56) Deprecated in Mac OS X v10.5
Gets a part used by the generic music component.

MusicGenericSetResourceNumbers (page 56) Deprecated in Mac OS X v10.5
Undocumented

MusicGetDrumNames (page 59) Deprecated in Mac OS X v10.5
Undocumented

MusicGetInfoText (page 60) Deprecated in Mac OS X v10.5
Undocumented

MusicGetInstrumentNames (page 62) Deprecated in Mac OS X v10.5
Undocumented

NewMusicMIDISendUPP (page 110) Deprecated in Mac OS X v10.5
Allocates a Universal Procedure Pointer for the MusicMIDISendProc callback.

NewMusicOfflineDataUPP (page 111) Deprecated in Mac OS X v10.5
Allocates a Universal Procedure Pointer for the MusicOfflineDataProc callback.

`NewTuneCallBackUPP` (page 111) Deprecated in Mac OS X v10.5
Allocates a Universal Procedure Pointer for the TuneCallBackProc callback.

`NewTunePlayCallBackUPP` (page 112) Deprecated in Mac OS X v10.5
Allocates a Universal Procedure Pointer for the TunePlayCallBackProc callback.

Functions

Callbacks

MusicMIDISendProc

Undocumented

```
typedef ComponentResult (*MusicMIDISendProcPtr) (ComponentInstance self, long
refCon, MusicMIDIPacket *mmp);
```

If you name your function `MyMusicMIDISendProc`, you would declare it this way:

```
ComponentResult MyMusicMIDISendProc (
    ComponentInstance    self,
    long                 refCon,
    MusicMIDIPacket      *mmp );
```

Parameters

`self`
Undocumented

`refCon`
A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

`mmp`
A pointer to a `MusicMIDIPacket` structure.

Return Value
See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In
`QuickTimeMusic.h`

MusicOfflineDataProc

Undocumented

```
typedef ComponentResult (*MusicOfflineDataProcPtr) (Ptr SoundData, long numBytes,
 long myRefCon);
```

If you name your function `MyMusicOfflineDataProc`, you would declare it this way:

```
ComponentResult MyMusicOfflineDataProc (
```

```
    Ptr     SoundData,
    long    numBytes,
    long    myRefCon );
```

Parameters

SoundData

Undocumented

numBytes

Undocumented

myRefCon

Undocumented

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeMusic.h`

TuneCallBackProc

Called when a sequence of music events is placed into a queue to be played.

```
typedef void (*TuneCallBackProcPtr) (const TuneStatus *status, long refCon);
```

If you name your function `MyTuneCallBackProc`, you would declare it this way:

```
void MyTuneCallBackProc (
    const TuneStatus    *status,
    long                refCon );
```

Parameters

status

A pointer to a `TuneStatus` structure.

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Declared In

`QuickTimeMusic.h`

TunePlayCallBackProc

Supports the TuneSetNoteChannels function.

```
typedef void (*TunePlayCallBackProcPtr) (unsigned long *event, long seed, long
refCon);
```

If you name your function `MyTunePlayCallBackProc`, you would declare it this way:

```
void MyTunePlayCallBackProc (
    unsigned long    *event,
    long             seed,
```

```
    long            refCon );
```

Parameters

event

A pointer to a QuickTime music event structure in the sequence data.

seed

A 32-bit value that is guaranteed to be different for each call to the callback routine (unless 2^32 calls are made, after which the values repeat), with one exception: the value passed at the beginning of a note is also passed at the end of the note's duration, together with a note structure or an extended note in which the velocity bits are set to 0.

refCon

A reference constant that the client code supplies to the callback.

Declared In
`QuickTimeMusic.h`

Data Types

AtomicInstrument

Represents a type used by the Music Architecture API.

```
typedef Handle AtomicInstrument;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

AtomicInstrumentPtr

Represents a type used by the Music Architecture API.

```
typedef Ptr AtomicInstrumentPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

GCPart

Defines a part in the QuickTime Music Architecture.

```
struct GCPart {
     long                hwInstrumentNumber;
     short               controller[128];
     long                volume;
     long                polyphony;
     long                midiChannel;
     GCInstrumentData    id;
 };
```

Fields

`hwInstrumentNumber`

Discussion
The instrument number of the instrument for the part.

`controller`

Discussion
An array of 128 bits identifying the available controllers; see `Music Controllers`. Bits are numbered from 1 to 128, starting with the most significant bit of the long word and continuing to the least significant of the last bit.

`volume`

Discussion
The sound volume for this part, ranging from -1.0 to +1.0. The high-order 8 bits contain the integer part; the low-order 8 bits contain the fractional part. A value of +1.0 constitutes the maximum volume of the user's computer. Negative values are silent but retain the magnitude of the volume setting.

`polyphony`

Discussion
The maximum number of voices.

`midiChannel`

Discussion
The system MIDI channel or, for a hardware device, the slot number.

`id`

Discussion
A `GCInstrumentData` structure.

Related Functions

`MusicDerivedSetInstrument` (page 48)
`MusicDerivedSetKnob` (page 49)
`MusicDerivedSetPart` (page 51)
`MusicDerivedSetPartInstrumentNumber` (page 51)
`MusicDerivedStorePartInstrument` (page 52)
`MusicGenericGetPart` (page 56)

Declared In
`QuickTimeMusic.h`

GenericKnobDescription

Describes a knob for the generic music component.

```
struct GenericKnobDescription {
     KnobDescription    kd;
     long               hw1;
     long               hw2;
     long               hw3;
     long               settingsID;
 };
```

Fields
`kd`

Discussion
A `KnobDescription` structure.

`hw1`

Discussion
Undocumented

`hw2`

Discussion
Undocumented

`hw3`

Discussion
Undocumented

`settingsID`

Discussion
Undocumented

Discussion
Undocumented

Related Functions
`MusicDerivedSetKnob` (page 49)

Declared In
`QuickTimeMusic.h`

GenericKnobDescriptionListHandle

Represents a type used by the Music Architecture API.

```
typedef GenericKnobDescriptionListPtr * GenericKnobDescriptionListHandle;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

GenericKnobDescriptionListPtr

Represents a type used by the Music Architecture API.

```
typedef GenericKnobDescriptionList * GenericKnobDescriptionListPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeMusic.h

InstrumentAboutInfo

Contains the information that appears in an instrument's About box and is returned by MusicGetInstrumentAboutInfo.

```
struct InstrumentAboutInfo {
     PicHandle    p;
     Str255       author;
     Str255       copyright;
     Str255       other;
 };
```

Fields
p

Discussion
A handle to a graphic for the About box.

author

Discussion
The author's name.

copyright

Discussion
The copyright information.

other

Discussion
Any other textual information.

Related Functions
MusicGetInstrumentAboutInfo (page 60)

Declared In
QuickTimeMusic.h

InstrumentInfoListHandle

Represents a type used by the Music Architecture API.

```
typedef InstrumentInfoListPtr * InstrumentInfoListHandle;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeMusic.h

InstrumentInfoListPtr

Represents a type used by the Music Architecture API.

```
typedef InstrumentInfoList * InstrumentInfoListPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

KnobDescription

Contains sound parameter values for a single knob.

```
struct KnobDescription {
     Str63    name;
     long     lowValue;
     long     highValue;
     long     defaultValue;
     long     flags;
     long     knobID;
 };
```

Fields
`name`

Discussion
The name of the knob.

`lowValue`

Discussion
The lowest number you can set the knob to.

`highValue`

Discussion
The highest number you can set the knob to.

`defaultValue`

Discussion
A value to use for the default. A default instrument is made of all default values.

`flags`

Discussion
Constants (see below) that provide various items of information about the knob. See these constants:

- `kKnobReadOnly`
- `kKnobInterruptUnsafe`
- `kKnobKeyrangeOverride`
- `kKnobGroupStart`
- `kKnobFixedPoint8`
- `kKnobFixedPoint16`
- `kKnobTypeNumber`
- `kKnobTypeGroupName`
- `kKnobTypeBoolean`
- `kKnobTypeNote`
- `kKnobTypePan`
- `kKnobTypeInstrument`
- `kKnobTypeSetting`
- `kKnobTypeMilliseconds`
- `kKnobTypePercentage`
- `kKnobTypeHertz`
- `kKnobTypeButton`

`knobID`

Discussion
A knob ID or index. A nonzero value in the high byte indicates that it is an ID. The knob index ranges from 1 to the number of knobs; the ID is an arbitrary number. Use the knob ID to refer to the knob in preference to the knob index, which may change.

Related Functions
`MusicGetDrumKnobDescription` (page 58)
`MusicGetInstrumentKnobDescription` (page 62)
`MusicGetKnobDescription` (page 64)

Declared In
`QuickTimeMusic.h`

MusicComponent

Represents a type used by the Music Architecture API.

```
typedef ComponentInstance MusicComponent;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

MusicController

Represents a type used by the Music Architecture API.

```
typedef SInt32 MusicController;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeMusic.h

MusicMIDIPacket

Describes MIDI data passed by note allocation calls.

```
struct MusicMIDIPacket {
     unsigned short    length;
     unsigned long     reserved;
     UInt8             data[249];
 };
```

Fields
length

Discussion
The length of the data in the packet.

reserved

Discussion
Contains 0, or one of the music packet status constants (see below). See these constants:

kMusicPacketPortLost

kMusicPacketPortFound

kMusicPacketTimeGap

data

Discussion
MIDI data.

Related Functions
MusicDerivedMIDISend (page 47)
MusicMIDIReadHookProc
MusicMIDISendProc
MusicSendMIDI (page 73)
NASendMIDI (page 101)
QTMIDISendMIDI (page 113)

Declared In
QuickTimeMusic.h

MusicMIDISendUPP

Represents a type used by the Music Architecture API.

```
typedef STACK_UPP_TYPE(MusicMIDISendProcPtr) MusicMIDISendUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

MusicOfflineDataUPP

Represents a type used by the Music Architecture API.

```
typedef STACK_UPP_TYPE(MusicOfflineDataProcPtr) MusicOfflineDataUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

NoteAllocator

Represents a type used by the Music Architecture API.

```
typedef ComponentInstance NoteAllocator;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

NoteChannel

Represents a type used by the Music Architecture API.

```
typedef struct OpaqueNoteChannel * NoteChannel;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

NoteRequest

Provides complete information for allocating a note channel.

```
struct NoteRequest {
     NoteRequestInfo    info;
     ToneDescription    tone;
 };
```

Fields
`info`

Discussion
A `NoteRequestInfo` structure.

`tone`

Discussion
A `ToneDescription` structure.

Related Functions
`NAGetNoteRequest` (page 91)
`NANewNoteChannel` (page 93)

Declared In
`QuickTimeMusic.h`

QTMIDIComponent

Represents a type used by the Music Architecture API.

```
typedef ComponentInstance QTMIDIComponent;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

QTMIDIPortListHandle

Represents a type used by the Music Architecture API.

```
typedef QTMIDIPortListPtr * QTMIDIPortListHandle;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

QTMIDIPortListPtr

Represents a type used by the Music Architecture API.

```
typedef QTMIDIPortList * QTMIDIPortListPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

Str31

Represents a type used by the Music Architecture API.

```
typedef unsigned char Str31;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`IOMacOSTypes.h`

SynthesizerConnections

Describes how a MIDI device is connected to the user's computer.

```
struct SynthesizerConnections {
     OSType    clientID;
     OSType    inputPortID;
     OSType    outputPortID;
     long      midiChannel;
     long      flags;
     long      unique;
     long      reserved1;
     long      reserved2;
 };
```

Fields
`clientID`

Discussion
The client ID provided by the MIDI Manager, or `'OMS '` for an OMS port.

`inputPortID`

Discussion
The ID provided by the MIDI Manager or OMS for the port used to SEND to the MIDI synthesizer.

`outputPortID`

Discussion
The ID provided by the MIDI Manager or OMS for the port that RECEIVES from a keyboard or other control device.

`midiChannel`

Discussion
The system MIDI channel or, for a hardware device, the slot number.

`flags`

Discussion
Constants (see below) that provide information about the type of connection. See these constants:

```
kSynthesizerConnectionMMgr
kSynthesizerConnectionOMS
kSynthesizerConnectionQT
kSynthesizerConnectionFMS
```

`unique`

Discussion
A unique ID you can use instead of an index to identify the synthesizer to the note allocator.

`reserved1`

Discussion
Reserved. Set to 0.

`reserved2`

Discussion
Reserved. Set to 0.

Related Functions
`NAGetRegisteredMusicDevice` (page 91)
`NARegisterMusicDevice` (page 99)

Declared In
`QuickTimeMusic.h`

SynthesizerDescription

Contains information about a synthesizer.

```
struct SynthesizerDescription {
     OSType           synthesizerType;
     Str31            name;
     unsigned long    flags;
     unsigned long    voiceCount;
     unsigned long    partCount;
     unsigned long    instrumentCount;
     unsigned long    modifiableInstrumentCount;
     unsigned long    channelMask;
     unsigned long    drumPartCount;
     unsigned long    drumCount;
     unsigned long    modifiableDrumCount;
     unsigned long    drumChannelMask;
     unsigned long    outputCount;
     unsigned long    latency;
     unsigned long    controllers[4];
     unsigned long    gmInstruments[4];
     unsigned long    gmDrums[4];
 };
```

Fields

`synthesizerType`

Discussion
The synthesizer type. This is the same as the music component subtype.

`name`

Discussion
Text name of the synthesizer type.

`flags`

Discussion
Constants (see below) that provide information about how the synthesizer works. See these constants:

- `kSynthesizerDynamicVoice`
- `kSynthesizerUsesMIDIPort`
- `kSynthesizerMicrotone`
- `kSynthesizerHasSamples`
- `kSynthesizerMixedDrums`
- `kSynthesizerSoftware`
- `kSynthesizerHardware`
- `kSynthesizerDynamicChannel`
- `kSynthesizerHogsSystemChannel`
- `kSynthesizerSlowSetPart`
- `kSynthesizerOffline`
- `kSynthesizerGM`

`voiceCount`

Discussion
Maximum polyphony.

`partCount`

Discussion
Maximum multi-timbrality (and MIDI channels).

`instrumentCount`

Discussion
The number of built-in ROM instruments. This does not include General MIDI instruments.

`modifiableInstrumentCount`

Discussion
The number of slots available for saving user-modified instruments.

`channelMask`

Discussion
Which channels a MIDI device always uses for instruments. Set to 0xFFFF for all channels.

`drumPartCount`

Discussion
The maximum multi-timbrality of drum parts. For synthesizers where drum kits are separated from instruments.

`drumCount`

Discussion
The number of built-in ROM drum kits. This does not include General MIDI drum kits. For synthesizers where drum kits are separated from instruments.

`modifiableDrumCount`

Discussion
The number of slots available for saving user-modified drum kits. For MIDI synthesizers where drum kits are separated from instruments.

`drumChannelMask`

Discussion
Which channels a MIDI device always uses for drum kits. Set to FFFF for all channels.

`outputCount`

Discussion
The number of audio outputs. This is usually 2.

`latency`

Discussion
The response time in microseconds.

`controllers`

Discussion
An array of 128 bits identifying the available controllers; see `Music Controllers`. Bits are numbered from 1 to 128, starting with the most significant bit of the long word and continuing to the least significant of the last bit.

`gmInstruments`

Discussion
An array of 128 bits giving the available General MIDI instruments.

`gmDrums`

Discussion
An array of 128 bits giving the available General MIDI drum kits.

Related Functions
`MusicGetDescription` (page 57)

Declared In
`QuickTimeMusic.h`

TuneCallBackUPP

Represents a type used by the Music Architecture API.

```
typedef STACK_UPP_TYPE(TuneCallBackProcPtr) TuneCallBackUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

TunePlayCallBackUPP

Represents a type used by the Music Architecture API.

```
typedef STACK_UPP_TYPE(TunePlayCallBackProcPtr) TunePlayCallBackUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

TunePlayer

Represents a type used by the Music Architecture API.

```
typedef ComponentInstance TunePlayer;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeMusic.h`

TuneStatus

Provides information on the currently playing tune.

```
struct TuneStatus {
     unsigned long *   tune;
     unsigned long *   tunePtr;
     TimeValue         time;
     short             queueCount;
     short             queueSpots;
     TimeValue         queueTime;
     long              reserved[3];
 };
```

Fields

`tune`

Discussion
The currently playing tune.

`tunePtr`

Discussion
Current position within the playing tune.

`time`

Discussion
Current tune time.

`queueCount`

Discussion
Number of tunes queued up.

`queueSpots`

Discussion
Number of tunes that can be added to the queue.

`queueTime`

Discussion
Total amount of playing time represented by tunes in the queue. This value can be very inaccurate.

`reserved`

Discussion
Reserved; set to 0.

Related Functions

`TuneGetStatus` (page 117)

Declared In

`QuickTimeMusic.h`

Constants

Generic Music Constants

Constants that represent generic music types.

```
enum {
  kGenericMusicComponentSubtype = 'gene'
};
enum {
  kGenericMusicDoMIDI           = 1 << 0, /* implement normal MIDI messages for
note, controllers, and program changes 0-127 */
  kGenericMusicBank0            = 1 << 1, /* implement instrument bank changes on
 controller 0 */
  kGenericMusicBank32           = 1 << 2, /* implement instrument bank changes on
 controller 32 */
  kGenericMusicErsatzMIDI       = 1 << 3, /* construct MIDI packets, but send them
 to the derived component */
  kGenericMusicCallKnobs        = 1 << 4, /* call the derived component with special
 knob format call */
  kGenericMusicCallParts        = 1 << 5, /* call the derived component with special
 part format call */
  kGenericMusicCallInstrument   = 1 << 6, /* call MusicDerivedSetInstrument for
MusicSetInstrument calls */
  kGenericMusicCallNumber       = 1 << 7, /* call MusicDerivedSetPartInstrumentNumber
 for MusicSetPartInstrumentNumber calls, & don't send any CO or bank stuff */
  kGenericMusicCallROMInstrument = 1 << 8, /* call MusicSetInstrument for
MusicSetPartInstrumentNumber for "ROM" instruments, passing params from the ROMi
resource */
  kGenericMusicAllDefaults      = 1 << 9 /* indicates that when a new instrument
is recalled, all knobs are reset to DEFAULT settings. True for GS modules */
};
enum {
  kGenericMusicKnob             = 1,
  kGenericMusicInstrumentKnob   = 2,
  kGenericMusicDrumKnob         = 3,
  kGenericMusicGlobalController = 4
};
enum {
  kGenericMusicMiscLongFirst    = 0,
  kGenericMusicMiscLongVoiceCount = 1,
  kGenericMusicMiscLongPartCount = 2,
  kGenericMusicMiscLongModifiableInstrumentCount = 3,
  kGenericMusicMiscLongChannelMask = 4,
  kGenericMusicMiscLongDrumPartCount = 5,
  kGenericMusicMiscLongModifiableDrumCount = 6,
  kGenericMusicMiscLongDrumChannelMask = 7,
  kGenericMusicMiscLongOutputCount = 8,
  kGenericMusicMiscLongLatency  = 9,
  kGenericMusicMiscLongFlags    = 10,
  kGenericMusicMiscLongFirstGMHW = 11,  /* number to add to locate GM main
instruments */
  kGenericMusicMiscLongFirstGMDrumHW = 12, /* number to add to locate GM drumkits
 */
  kGenericMusicMiscLongFirstUserHW = 13, /* First hw number of user instruments
(presumed sequential) */
  kGenericMusicMiscLongLast     = 14
};
enum {
  kGenericMusicResFirst         = 0,
  kGenericMusicResMiscStringList = 1,   /* STR# 1: synth name, 2:about
author,3:aboutcopyright,4:aboutother */
  kGenericMusicResMiscLongList  = 2,    /* Long various params, see list below */
  kGenericMusicResInstrumentList = 3,   /* NmLs of names and shorts, categories
```

```
prefixed by two bullet characters */
  kGenericMusicResDrumList      = 4,    /* NmLs of names and shorts */
  kGenericMusicResInstrumentKnobDescriptionList = 5, /* Knob */
  kGenericMusicResDrumKnobDescriptionList = 6, /* Knob */
  kGenericMusicResKnobDescriptionList = 7, /* Knob */
  kGenericMusicResBitsLongList  = 8,    /* Long back to back bitmaps of controllers,
 gminstruments, and drums */
  kGenericMusicResModifiableInstrumentHW = 9, /* Shrt same as the hw shorts trailing
 the instrument names, a shortlist */
  kGenericMusicResGMTranslation = 10,   /* Long 128 long entries, 1 for each gm
inst, of local instrument numbers 1-n (not hw numbers) */
  kGenericMusicResROMInstrumentData = 11, /* knob lists for ROM instruments, so
the knob values may be known */
  kGenericMusicResAboutPICT     = 12,   /* picture for aboutlist. must be present
 for GetAbout call to work */
  kGenericMusicResLast          = 13
};
enum {
  kMusicGenericRange            = 0x0100,
  kMusicDerivedRange            = 0x0200
};
```

Constants

`kGenericMusicAllDefaults`

Indicates that when a new instrument is recalled, all knobs are reset to DEFAULT settings. True for GS modules.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicDrumKnob`

Value is 3.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicMiscLongFirstGMHW`

Number to add to locate GM main instruments.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicMiscLongFirstGMDrumHW`

Number to add to locate GM drumkits.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicMiscLongFirstUserHW`

First HW number of user instruments (presumed sequential).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResMiscStringList`

STR# 1: `synth` name, 2:about author,3:aboutcopyright,4:aboutother.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResMiscLongList`
Long various `params`, see list below.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResInstrumentList`
NmLs of names and shorts, categories prefixed by two bullet characters.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResDrumList`
NmLs of names and shorts.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResInstrumentKnobDescriptionList`
Knob.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResDrumKnobDescriptionList`
Knob.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResKnobDescriptionList`
Knob.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResBitsLongList`
Long back to back bitmaps of controllers, gminstruments, and drums.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResModifiableInstrumentHW`
Short same as the HW shorts trailing the instrument names, a short list.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResGMTranslation`
Long 128 long entries, 1 for each gm instrument, of local instrument numbers 1-n (not HW numbers).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResROMInstrumentData`
Knob lists for ROM instruments, so the knob values may be known.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kGenericMusicResAboutPICT`

`Picture` for about list. Must be present for `GetAbout` call to work.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

Declared In
`QuickTimeMusic.h`

MusicSetPartAtomicInstrument Values

Constants passed to MusicSetPartAtomicInstrument.

```
enum {
  kGetAtomicInstNoExpandedSamples = 1 << 0,
  kGetAtomicInstNoOriginalSamples = 1 << 1,
  kGetAtomicInstNoSamples       = kGetAtomicInstNoExpandedSamples |
kGetAtomicInstNoOriginalSamples,
  kGetAtomicInstNoKnobList      = 1 << 2,
  kGetAtomicInstNoInstrumentInfo = 1 << 3,
  kGetAtomicInstOriginalKnobList = 1 << 4,
  kGetAtomicInstAllKnobs        = 1 << 5 /* return even those that are set to
default*/
};
```

Declared In
`QuickTimeMusic.h`

MusicGetInstrumentInfo Values

Constants passed to MusicGetInstrumentInfo.

```
enum {
  kGetInstrumentInfoNoBuiltIn   = 1 << 0,
  kGetInstrumentInfoMidiUserInst = 1 << 1,
  kGetInstrumentInfoNoIText     = 1 << 2
};
```

Declared In
`QuickTimeMusic.h`

kInstrumentMatchGMNumber

Constants grouped with kInstrumentMatchGMNumber.

```
enum {
  kInstrumentMatchSynthesizerType = 1,
  kInstrumentMatchSynthesizerName = 2,
  kInstrumentMatchName          = 4,
  kInstrumentMatchNumber        = 8,
  kInstrumentMatchGMNumber      = 16,
  kInstrumentMatchGSNumber      = 32
};
```

Declared In
`QuickTimeMusic.h`

kKnobBasic

Constants grouped with kKnobBasic.

```
enum {
  kKnobBasic                    = 8,    /* knob shows up in certain simplified
lists of knobs */
  kKnobReadOnly                 = 16,   /* knob value cannot be changed by user or
 with a SetKnob call */
  kKnobInterruptUnsafe          = 32,   /* only alter this knob from foreground
task time (may access toolbox) */
  kKnobKeyrangeOverride         = 64,   /* knob can be overridden within a single
 keyrange (software synth only) */
  kKnobGroupStart               = 128,  /* knob is first in some logical group of
 knobs */
  kKnobFixedPoint8              = 1024,
  kKnobFixedPoint16             = 2048, /* One of these may be used at a time. */
  kKnobTypeNumber               = 0 << 12,
  kKnobTypeGroupName            = 1 << 12, /* "knob" is really a group name for
display purposes */
  kKnobTypeBoolean              = 2 << 12, /* if range is greater than 1, its a
multi-checkbox field */
  kKnobTypeNote                 = 3 << 12, /* knob range is equivalent to MIDI keys
 */
  kKnobTypePan                  = 4 << 12, /* range goes left/right (lose this? )
 */
  kKnobTypeInstrument           = 5 << 12, /* knob value = reference to another
instrument number */
  kKnobTypeSetting              = 6 << 12, /* knob value is 1 of n different things
 (eg, fm algorithms) popup menu */
  kKnobTypeMilliseconds         = 7 << 12, /* knob is a millisecond time range */
  kKnobTypePercentage           = 8 << 12, /* knob range is displayed as a Percentage
 */
  kKnobTypeHertz                = 9 << 12, /* knob represents frequency */
  kKnobTypeButton               = 10 << 12 /* momentary trigger push button */
};
```

Constants

`kKnobReadOnly`

The knob value cannot be changed by the user or with a set knob call.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobInterruptUnsafe`
Alter this knob only from foreground task time.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobKeyrangeOverride`
The knob can be overridden within a single key range (software synthesizer only).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobGroupStart`
The knob is first in some logical group of knobs.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobFixedPoint8`
Interpret knob numbers as fixed-point 8-bit.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobFixedPoint16`
Interpret knob numbers as fixed-point 16-bit.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeNumber`
The knob value is a numerical value.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeGroupName`
The name of the knob is really a group name for display purposes.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeBoolean`
The knob is an on/off knob. If the range of the knob (as specified by the low value and high value in the knob description structure) is greater than one, the knob is a multi-checkbox field.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeNote`
The knob value range is equivalent to MIDI keys.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypePan`
The knob value is the pan setting and is within a range (as specified by the low value and high value in the knob description structure) that goes from left to right.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeInstrument`
The knob value is a reference to another instrument number.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeSetting`
The knob value is one of several different discrete settings; for example, items on a pop-up menu.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeMilliseconds`
The knob value is in milliseconds.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypePercentage`
The knob value is a percentage of the range.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kKnobTypeHertz`
The knob value represents frequency.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

Declared In
`QuickTimeMusic.h`

MusicMIDIPacket Values

Constants passed to MusicMIDIPacket.

```
enum {
  kMusicPacketPortLost          = 1,    /* received when application loses the
default input port */
  kMusicPacketPortFound         = 2,    /* received when application gets it back
 out from under someone else's claim */
  kMusicPacketTimeGap          = 3     /* data[0] = number of milliseconds to keep
 the MIDI line silent */
};
```

Constants
`kMusicPacketPortLost`
The application has lost the default input port.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kMusicPacketPortFound`
The application has retrieved the input port from the previous owner.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

Declared In
QuickTimeMusic.h

kPickDontMix

Constants grouped with kPickDontMix.

```
enum {
  kPickDontMix                  = 1,    /* dont mix instruments with drum sounds
*/
  kPickSameSynth                = 2,    /* only allow the same device that went
in, to come out */
  kPickUserInsts               = 4,    /* show user insts in addition to ROM voices
 */
  kPickEditAllowEdit            = 8,    /* lets user switch over to edit mode */
  kPickEditAllowPick            = 16,   /* lets the user switch over to pick mode
 */
  kPickEditSynthGlobal          = 32,   /* edit the global knobs of the synth */
  kPickEditControllers          = 64    /* edit the controllers of the notechannel
 */
};
```

Declared In
QuickTimeMusic.h

kSetAtomicInstCallerGuarantees

Constants grouped with kSetAtomicInstCallerGuarantees.

```
enum {
  kSetAtomicInstKeepOriginalInstrument = 1 << 0,
  kSetAtomicInstShareAcrossParts = 1 << 1, /* inst disappears when app goes away*/
  kSetAtomicInstCallerTosses    = 1 << 2, /* the caller isn't keeping a copy around
 (for NASetAtomicInstrument)*/
  kSetAtomicInstCallerGuarantees = 1 << 3, /* the caller guarantees a copy is
around*/
  kSetAtomicInstInterruptSafe   = 1 << 4, /* dont move memory at this time (but
process at next task time)*/
  kSetAtomicInstDontPreprocess  = 1 << 7 /* perform no further preprocessing because
 either 1)you know the instrument is digitally clean, or 2) you got it from a
GetPartAtomic*/
};
```

Declared In
QuickTimeMusic.h

kSynthesizerConnectionFMS

Constants grouped with kSynthesizerConnectionFMS.

```
enum {
  kSynthesizerConnectionFMS     = 1,    /* this connection imported from FMS */
  kSynthesizerConnectionMMgr    = 2,    /* this connection imported from the MIDI
 Mgr */
  kSynthesizerConnectionOMS     = 4,    /* this connection imported from OMS */
  kSynthesizerConnectionQT      = 8,    /* this connection is a QuickTime-only port
 */
  kSynthesizerConnectionOSXMIDI = 16,   /* this connection is an OS X CoreMIDI port
 */
                                        /* lowest five bits are mutually exclusive;
 combinations reserved for future use.*/
  kSynthesizerConnectionUnavailable = 256 /* port exists, but cannot be used just
 now */
};
```

Constants

`kSynthesizerConnectionFMS`

This connection is imported from the FreeMIDI system.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerConnectionMMgr`

This connection is imported from the MIDI Manager.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerConnectionOMS`

This connection is imported from the Open Music System (OMS).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerConnectionQT`

This connection is a QuickTime-only port.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

Declared In

`QuickTimeMusic.h`

kSynthesizerDLS

Constants grouped with kSynthesizerDLS.

```
enum {
  kSynthesizerDynamicVoice      = 1 << 0, /* can assign voices on the fly (else,
polyphony is very important */
  kSynthesizerUsesMIDIPort      = 1 << 1, /* must be patched through MIDI Manager
 */
  kSynthesizerMicrotone         = 1 << 2, /* can play microtonal scales */
  kSynthesizerHasSamples        = 1 << 3, /* synthesizer has some use for sampled
 data */
  kSynthesizerMixedDrums        = 1 << 4, /* any part can play drum parts, total
= instrument parts */
  kSynthesizerSoftware          = 1 << 5, /* implemented in main CPU software ==
uses cpu cycles */
  kSynthesizerHardware          = 1 << 6, /* is a hardware device (such as nubus,
 or maybe DSP?) */
  kSynthesizerDynamicChannel    = 1 << 7, /* can move any part to any channel or
disable each part. (else we assume it lives on all channels in masks) */
  kSynthesizerHogsSystemChannel = 1 << 8, /* can be channelwise dynamic, but always
 responds on its system channel */
  kSynthesizerHasSystemChannel  = 1 << 9, /* has some "system channel" notion to
distinguish it from multiple instances of the same device (GM devices dont) */
  kSynthesizerSlowSetPart      = 1 << 10, /* SetPart() and SetPartInstrumentNumber()
 calls do not have rapid response, may glitch notes */
  kSynthesizerOffline           = 1 << 12, /* can enter an offline synthesis mode
 */
  kSynthesizerGM                = 1 << 14, /* synth is a GM device */
  kSynthesizerDLS               = 1 << 15, /* synth supports DLS level 1 */
  kSynthesizerSoundLocalization = 1 << 16 /* synth supports extremely baroque,
nonstandard, and proprietary "apple game sprockets" localization parameter set */
};
```

Constants

`kSynthesizerDynamicVoice`

Voices can be assigned to parts on the fly with this synthesizer (otherwise, polyphony is very important).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerUsesMIDIPort`

This synthesizer must be patched through a MIDI system, such as the MIDI Manager or OMS.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerMicrotone`

This synthesizer can play microtonal scales.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerHasSamples`

This synthesizer has some use for sampled audio data.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerMixedDrums`

Any part of this synthesizer can play drum parts.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeMusic.h`.

`kSynthesizerSoftware`
: This synthesizer is implemented in main CPU software and uses CPU cycles.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerHardware`
: This synthesizer is a hardware device, not a software synthesizer or MIDI device.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerDynamicChannel`
: This synthesizer can move any part to any channel or disable each part. For devices only.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerHogsSystemChannel`
: Even if the `kSynthesizerDynamicChannel` bit is set, this synthesizer always responds on its system channel. For MIDI devices only.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerSlowSetPart`
: This synthesizer does not respond rapidly to the various set part and set part instrument calls.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerOffline`
: This synthesizer can enter an offline synthesis mode.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

`kSynthesizerGM`
: This synthesizer is a General MIDI device.

  Available in Mac OS X v10.0 and later.

  Declared in `QuickTimeMusic.h`.

Declared In
`QuickTimeMusic.h`

TuneSetPartMix Values

Constants passed to TuneSetPartMix.

```
enum {
  kTuneMixMute                  = 1,    /* disable a part */
  kTuneMixSolo                  = 2     /* if any parts soloed, play only soloed
parts */
};
```

Declared In
`QuickTimeMusic.h`

kTuneDontClipNotes

Constants grouped with kTuneDontClipNotes.

```
enum {
  kTuneStartNow                 = 1,    /* start after buffer is implied */
  kTuneDontClipNotes            = 2,    /* allow notes to finish their durations
outside sample */
  kTuneExcludeEdgeNotes         = 4,    /* dont play notes that start at end of
tune */
  kTuneQuickStart               = 8,    /* Leave all the controllers where they
are, ignore start time */
  kTuneLoopUntil                = 16,   /* loop a queued tune if there's nothing
else in the queue*/
  kTunePlayDifference           = 32,   /* by default, the tune difference is
skipped*/
  kTunePlayConcurrent           = 64,   /* dont block the next tune sequence with
 this one*/
  kTuneStartNewMaster           = 16384
};
```

Declared In

QuickTimeMusic.h

Deprecated QuickTime Music Architecture Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

DisposeMusicMIDISendUPP

Disposes of a MusicMIDISendUPP pointer. (Deprecated in Mac OS X v10.5.)

```
void DisposeMusicMIDISendUPP (
   MusicMIDISendUPP userUPP
);
```

Parameters

userUPP

A `MusicMIDISendUPP` pointer. See `Universal Procedure Pointers`.

Return Value

You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

DisposeMusicOfflineDataUPP

Disposes of a MusicOfflineDataUPP pointer. (Deprecated in Mac OS X v10.5.)

```
void DisposeMusicOfflineDataUPP (
   MusicOfflineDataUPP userUPP
);
```

Parameters

userUPP

A `MusicOfflineDataUPP` pointer. See `Universal Procedure Pointers`.

Return Value

You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

DisposeTuneCallBackUPP

Disposes of a TuneCallBackUPP pointer. (Deprecated in Mac OS X v10.5.)

```
void DisposeTuneCallBackUPP (
   TuneCallBackUPP userUPP
);
```

Parameters

userUPP

A `TuneCallBackUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

DisposeTunePlayCallBackUPP

Disposes of a TunePlayCallBackUPP pointer. (Deprecated in Mac OS X v10.5.)

```
void DisposeTunePlayCallBackUPP (
   TunePlayCallBackUPP userUPP
);
```

Parameters

userUPP

A `TunePlayCallBackUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicDerivedCloseResFile

Closes a music movie resource file. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedCloseResFile (
   MusicComponent mc,
   short resRefNum
);
```

Parameters

mc

A music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

resRefNum

The resource file to be closed. Your application obtains this value from the `OpenMovieFile` function.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicDerivedMIDISend

Sends a MIDI packet to a music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedMIDISend (
   MusicComponent mc,
   MusicMIDIPacket *packet
);
```

Parameters

mc

A music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent` function.

packet

A pointer to the music MIDI packet to be sent.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedOpenResFile

Opens the music resource file for a music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedOpenResFile (
   MusicComponent mc
);
```

Parameters

mc

A music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedSetInstrument

The complete instrument defined by the Part structure to the synthesizer. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedSetInstrument (
   MusicComponent mc,
   long partNumber,
   GCPart *p
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

partNumber

The number of the part for this operation.

p
A pointer to the part for this operation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedSetKnob

Called when any of the synthesizer's knobs are altered. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedSetKnob (
   MusicComponent mc,
   long knobType,
   long knobNumber,
   long knobValue,
   long partNumber,
   GCPart *p,
   GenericKnobDescription *gkd
);
```

Parameters

mc
The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

knobType
The type of knob that has been altered (see below). See these constants:
`kGenericMusicKnob`
`kGenericMusicInstrumentKnob`
`kGenericMusicDrumKnob`

knobNumber
The number of the knob that has been altered.

knobValue
The new value of the altered knob.

partNumber
The number of the part whose knob has been altered.

p
A pointer to the part whose knob has been altered.

gkd
A `GenericKnobDescription` structure for the knob.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is called when any knob on the synthesizer is altered. It should look at the `GCPart` and the `GenericKnobDescription` structures and address the synthesizer hardware appropriately to set the new knob value. For a MIDI device, this means to construct a system-exclusive MIDI packet and send it to the MIDI routine received by the `MusicDerivedSetMIDI` (page 50) call.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedSetMIDI

Sets the MIDI channel and other MIDI settings for MIDI output only. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedSetMIDI (
   MusicComponent mc,
   MusicMIDISendUPP midiProc,
   long refcon,
   long midiChannel
);
```

Parameters

mc
The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

midiProc
A pointer to the `MusicMIDISendProc` callback in your music component for performing MIDI output.

refcon
A reference constant sent to the callback specified by the `midiProc` parameter. Use this parameter to point to a data structure containing any information your callback needs.

midiChannel
The MIDI channel to use for the operation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
A derived component for a MIDI synthesizer receives this call soon after it is opened. It should store the `midiProc`, `refCon`, and `midiChannel` parameters in its global variables. When the derived component needs to communicate with the synthesizer, it calls your `MusicMIDISendProc` function with this reference constant. The `midiChannel` variable specifies the "system channel" of the device.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedSetPart

Sets the polyphony for the part specified in the GCPart structure. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedSetPart (
   MusicComponent mc,
   long partNumber,
   GCPart *p
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

partNumber

The number of the part for this operation.

p

A pointer to the part for this operation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedSetPartInstrumentNumber

Sets the instrument specified in the GCPart structure. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedSetPartInstrumentNumber (
   MusicComponent mc,
   long partNumber,
   GCPart *p
);
```

Parameters

mc

A music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

partNumber

The number of the part for this operation.

p

A pointer to the part for this operation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicDerivedStorePartInstrument

Undocumented (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicDerivedStorePartInstrument (
   MusicComponent mc,
   long partNumber,
   GCPart *p,
   long instrumentNumber
);
```

Parameters

mc

An instance of the music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

partNumber

The number of the part for this operation.

p

A pointer to the part for this operation.

instrumentNumber

Number of the instrument for this part. You can use `MusicFindTone` (page 53) to get an instrument number.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicFindTone

Returns the number of the best-matching instrument provided by a specified music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicFindTone (
   MusicComponent mc,
   ToneDescription *td,
   long *libraryIndexOut,
   unsigned long *fit
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

td
Pointer to a `ToneDescription` structure.

libraryIndexOut
On return, contains the number of the best-matching instrument. Only General MIDI numbers are guaranteed to be the same for later instantiations of the component.

fit
On return, a constant (see below) that indicates how well an instrument matches the tone description. See these constants:

`kInstrumentMatchSynthesizerType`
`kInstrumentMatchSynthesizerName`
`kInstrumentMatchName`
`kInstrumentMatchNumber`
`kInstrumentMatchGMNumber`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The music component searches for an instrument as follows:

If the `synthesizerType` field of the `td` parameter matches the type of the specified music component, it first tries to find an instrument that matches the value of the `instrumentNumber` field of the `td` parameter. If this value is in the range 129-16512, which specifies a GS instrument, and the GS instrument is not available, it tries to find the General MIDI instrument that corresponds to it, which has the number `((GSinstrumentnumber - 1) & 0x7F)` + 1. If the value is greater than 16512, which specifies a transient

ROM instrument or internal instrument index value, it tries to find an instrument that matches the `synthesizerName` field of the `td` parameter. If that fails, it tries to find an instrument that matches the value of the `gmNumber` field of the `td` parameter.

If the `synthesizerType` field of the `td` parameter does not match the type of the specified music component, it tries to find an instrument that matches the value of the `gmNumber` field of the `td` parameter.

If none of these rules apply, or the fields are blank (0 for the `type` or numeric fields, or zero-length for the strings), then the call returns instrument 1 and a fit parameter of zero.

The `synthesizerName` field may be ignored by the component; it is used by the note allocator when deciding which music device to use.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicGenericConfigure

Informs the generic music component what services your music component requires and points to any resources that are necessary. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGenericConfigure (
   MusicComponent mc,
   long mode,
   long flags,
   long baseResID
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling the Component Manager's `OpenComponent` or `OpenDefaultComponent` function.

mode

Must be 0.

flags

Flags (see below) that control the importation of MIDI files. See these constants:

- kGenericMusicDoMIDI
- kGenericMusicBank0
- kGenericMusicBank32
- kGenericMusicErsatzMIDI
- kGenericMusicCallKnobs
- kGenericMusicCallParts
- kGenericMusicCallInstrument
- kGenericMusicCallNumber
- kGenericMusicCallROMInstrument

baseResID

The resource ID of the lowest-numbered resource used by your music component.

Return Value

See Error Codes. Returns noErr if there is no error.

Discussion

The baseResID parameter is the lowest resource ID used by your component for the standard resources described above. Since the resource numbers are relative to this, you can include several music components in a single system extension.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

QuickTimeMusic.h

MusicGenericGetKnobList

Gets a list of the knobs of a given type for the generic music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGenericGetKnobList (
   MusicComponent mc,
   long knobType,
   GenericKnobDescriptionListHandle *gkdlH
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling OpenComponent or OpenDefaultComponent.

knobType

A constant (see below) that defines the type of knob. See these constants:

- kGenericMusicKnob
- kGenericMusicInstrumentKnob
- kGenericMusicDrumKnob

gkdlH

On return, a pointer to a handle to a `GenericKnobDescriptionList` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGenericGetPart

Gets a part used by the generic music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGenericGetPart (
   MusicComponent mc,
   long partNumber,
   GCPart **part
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

partNumber

The number of the part for this operation.

part

A handle to a `GCPart` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGenericSetResourceNumbers

Undocumented (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGenericSetResourceNumbers (
   MusicComponent mc,
   Handle resourceIDH
);
```

Parameters

mc

The instance of the generic music component. Your software obtains this reference when calling `OpenComponent` or `OpenDefaultComponent`.

resourceIDH

A handle to a resource ID.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetDescription

Returns a structure describing the synthesizer controlled by the music component device. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetDescription (
   MusicComponent mc,
   SynthesizerDescription *sd
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

sd

Pointer to a `SynthesizerDescription` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
QuickTimeMusic.h

MusicGetDeviceConnection

Determines how many hardware synthesizers are available to a music component and gets the IDs for those devices. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetDeviceConnection (
   MusicComponent mc,
   long index,
   long *id1,
   long *id2
);
```

Parameters

mc
Music component returned by NAGetRegisteredMusicDevice (page 91).

index
Index of the device for which you want to find out the IDs. Set to 0 if you are calling to get the number of hardware devices.

id1
On return, a hardware synthesizer ID.

id2
On return, another hardware synthesizer ID.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
To get the number of hardware synthesizers available to the music component specified in the mc parameter and an index you can use to request ID numbers for a specific device, call this function with a value of 0 for the index parameter. You can then pass an index value in the index parameter, and the function returns hardware synthesizer IDs in the id1 and id2 parameters.

Special Considerations

This function is implemented only for hardware synthesizers, such as PCI card devices.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetDrumKnobDescription

Returns a description of a drum kit knob. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetDrumKnobDescription (
   MusicComponent mc,
   long knobIndex,
   KnobDescription *mkd
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

knobIndex

A knob index or knob ID.

mkd

A pointer to a `KnobDescription` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetDrumNames

Undocumented (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetDrumNames (
   MusicComponent mc,
   long modifiableInstruments,
   Handle *instrumentNumbers,
   Handle *instrumentNames
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

modifiableInstruments

Undocumented

instrumentNumbers

Undocumented

instrumentNames

A pointer to a handle to the requested list of instrument name strings, formatted as a short integer followed by packed strings.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetInfoText

Undocumented (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetInfoText (
   MusicComponent mc,
   long selector,
   Handle *textH,
   Handle *styleH
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

selector
Undocumented

textH
Undocumented

styleH
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetInstrumentAboutInfo

Obtains the information about an instrument that appears in its About box. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetInstrumentAboutInfo (
   MusicComponent mc,
   long part,
   InstrumentAboutInfo *iai
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

Number of the part containing the instrument for which you want information.

iai

On return, a pointer to an `InstrumentAboutInfo` structure for the instrument currently on the specified synthesizer part.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicGetInstrumentInfo

Obtains a list of instruments supported by a synthesizer. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetInstrumentInfo (
   MusicComponent mc,
   long getInstrumentInfoFlags,
   InstrumentInfoListHandle *infoListH
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

getInstrumentInfoFlags

Flags (see below) that specify limits to the list of instruments. See these constants:

`kGetInstrumentInfoNoBuiltIn`

`kGetInstrumentInfoMidiUserInst`

`kGetInstrumentInfoNoIText`

infoListH

On return, a pointer to a handle to an `InstrumentInfoList` structure that contains the list of instruments. This handle must be disposed of by the caller.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetInstrumentKnobDescription

Obtains the description of an instrument knob. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetInstrumentKnobDescription (
   MusicComponent mc,
   long knobIndex,
   KnobDescription *mkd
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

knobIndex
A knob index or knob ID.

mkd
On return, a `KnobDescription` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicGetInstrumentNames

Undocumented (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetInstrumentNames (
   MusicComponent mc,
   long modifiableInstruments,
   Handle *instrumentNames,
   Handle *instrumentCategoryLasts,
   Handle *instrumentCategoryNames
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

modifiableInstruments

Undocumented

instrumentNames

A pointer to a handle to the requested list of instrument name strings, formatted as a short integer followed by packed strings.

instrumentCategoryLasts

A pointer to a handle to a group of short integers, the first of which contains the number of integers to follow.

instrumentCategoryNames

A pointer to a handle to the requested list of instrument category name strings, formatted as a short integer followed by packed strings.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicGetKnob

Returns the value of the specified global synthesizer knob. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetKnob (
   MusicComponent mc,
   long knobID
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

knobID

Knob index or ID.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
A global knob controls an aspect of the entire synthesizer. It is not specific to a part within the synthesizer.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicGetKnobDescription

Returns a pointer to an initialized knob description structure describing a global synthesizer knob. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetKnobDescription (
   MusicComponent mc,
   long knobIndex,
   KnobDescription *mkd
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

knobIndex
Knob index or ID.

mkd
Pointer to a `KnobDescription` structure. The initialized structure provides default values associated with the particular knob.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
A global knob controls an aspect of the entire synthesizer; it is not limited to a part within the synthesizer. You can use the information returned by a call to this function to reset a knob to some known, usable value.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
QuickTimeMusic.h

MusicGetKnobSettingStrings

Returns a list of knob setting names known by the specified music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetKnobSettingStrings (
   MusicComponent mc,
   long knobIndex,
   long isGlobal,
   Handle *settingsNames,
   Handle *settingsCategoryLasts,
   Handle *settingsCategoryNames
);
```

Parameters

mc
Music component instance identifier returned by NAGetRegisteredMusicDevice (page 91).

knobIndex
The knob index or knob ID.

isGlobal
If a knob index is used, indicates whether the specified knob is a global knob.

settingsNames
The requested list of knob setting strings formatted as a short followed by packed strings.

settingsCategoryLasts
A group of short integers, the first of which contains the number of shorts to follow.

settingsCategoryNames
Knob setting category names formatted as a short integer followed by a list of names.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
All handles must be disposed of by the caller.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetMasterTune

Returns the synthesizer's master tuning as a fixed-point value in semitones. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetMasterTune (
   MusicComponent mc
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

Return Value
A fixed-point value representing the synthesizer's master tuning. The value is a fixed 16.16 number, allowing shifts by fractional values.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicGetMIDIPorts

Returns the number of input and output ports a MIDI device has. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetMIDIPorts (
   MusicComponent mc,
   long *inputPortCount,
   long *outputPortCount
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

inputPortCount

On return, the number of input MIDI ports available to the music component.

outputPortCount

On return, the number of output MIDI ports available to the music component.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This call is implemented only for hardware synthesizers, such as PCI card devices.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetMIDIProc

Returns a pointer to the procedure a music component is using to process external MIDI notes. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetMIDIProc (
   MusicComponent mc,
   MusicMIDISendUPP *midiSendProc,
   long *refCon
);
```

Parameters

mc

Music component instance identifier returned by NAGetRegisteredMusicDevice (page 91).

midiSendProc

Pointer to a MIDI serial port MusicMIDISendProc callback that processes external MIDI notes. This function was set by a previous call to MusicSetMIDIProc (page 75). If no function has been set with MusicSetMIDIProc, this parameter returns 0.

refCon

A reference constant. The Movie Toolbox passes this reference constant to your MusicMIDISendProc each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetPart

Returns the MIDI channel and maximum polyphony for a particular part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPart (
   MusicComponent mc,
   long part,
   long *midiChannel,
   long *polyphony
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

The music component part requested.

midiChannel

On return, a pointer to a MIDI channel. For non-MIDI devices, the MIDI channel pointed to by this parameter is 0.

polyphony

On return, a pointer to the maximum number of voices or polyphony for the part..

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicGetPartAtomicInstrument

Returns the atomic instrument currently in a part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPartAtomicInstrument (
   MusicComponent mc,
   long part,
   AtomicInstrument *ai,
   long flags
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

The part with the atomic instrument.

ai

On return, an atomic instrument.

flags

A constant (see below) that specifies what pieces of information about an atomic instrument the caller is interested in. See these constants:

```
kGetAtomicInstNoExpandedSamples
kGetAtomicInstNoOriginalSamples
kGetAtomicInstNoSamples
kGetAtomicInstNoKnobList
kGetAtomicInstNoInstrumentInfo
kGetAtomicInstOriginalKnobList
kGetAtomicInstAllKnobs
```

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicGetPartController

Returns the value of a specified controller on a specified part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPartController (
   MusicComponent mc,
   long part,
   MusicController controllerNumber
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

Part whose controller value you want to get.

controllerNumber

On return, the controller number; see `Music Controllers`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetPartInstrumentNumber

Returns the instrument number currently assigned to a part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPartInstrumentNumber (
   MusicComponent mc,
   long part
);
```

Parameters

mc

Music component instance identifier returned by NAGetRegisteredMusicDevice (page 91).

part

Part number containing the instrument.

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicGetPartKnob

Retrieves the current value of a knob for a part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPartKnob (
   MusicComponent mc,
   long part,
   long knobID
);
```

Parameters

mc

Music component instance identifier returned by NAGetRegisteredMusicDevice (page 91).

part

The part number.

knobID

The knob index or ID.

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicGetPartName

Returns the string name of a part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicGetPartName (
   MusicComponent mc,
   long part,
   StringPtr name
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part
Part to get the name of.

name
On return, a string containing the part name.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The name string is used by selection dialog boxes or configuration information.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicPlayNote

Plays a note on a specified part at a specified pitch and velocity. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicPlayNote (
   MusicComponent mc,
   long part,
   long pitch,
   long velocity
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

The part to play the note on.

pitch

The pitch at which to play the note. If the pitch is specified by a number from 0 to 127, it is a MIDI pitch, where 60 is middle C. If the pitch is a positive number above 65535, the value is a fixed-point pitch value. Thus, microtonal values may be specified.

velocity

How hard to strike the key. Values are 0-127 where 0 is silence. Velocity refers to how hard the key is struck (if performed on a keyboard-instrument); typically, this translates directly to volume, but on many synthesizers this also subtly alters the timbre of the tone.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The current note continues to play until a `MusicPlayNote` function with the same pitch and velocity of 0 turns the note off.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

QTMusicToo

Declared In

`QuickTimeMusic.h`

MusicResetPart

Silences all sounds on a specified part and resets all controllers on that part to their default values. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicResetPart (
   MusicComponent mc,
   long part
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

The number of the part.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The default value to which controllers on the part are set is 0 for all controllers except volume. Volume is set to its maximum, 32767 (0x7FFF).

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicSendMIDI

Sends a MIDI packet to a specified port. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSendMIDI (
   MusicComponent mc,
   long portIndex,
   MusicMIDIPacket *mp
);
```

Parameters

mc

Music component instance returned by `NAGetRegisteredMusicDevice` (page 91).

portIndex

The index of the port to send the MIDI packet to. The index value is 1 through the port count returned by `MusicGetMIDIPorts` (page 66).

mp

A pointer to the music MIDI packet to be sent. The function sends the MIDI music packet specified by this parameter to the specified port.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This call is implemented only for hardware synthesizers, such as PCI card devices.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicSetKnob

Modifies the value of the specified global synthesizer knob. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetKnob (
   MusicComponent mc,
   long knobID,
   long knobValue
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

knobID
Knob index or ID.

knobValue
Value for specified knob.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
A global knob controls an aspect of the entire synthesizer; it is not limited to a part within the synthesizer.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicSetMasterTune

Alters a synthesizer's master tuning. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetMasterTune (
   MusicComponent mc,
   long masterTune
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

masterTune

The amount by which to transpose the entire synthesizer in pitch. The value is a fixed 16.16 number, allowing shifts by fractional values.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicSetMIDIProc

Informs the music component what procedure to call when it needs to send MIDI data. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetMIDIProc (
   MusicComponent mc,
   MusicMIDISendUPP midiSendProc,
   long refCon
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

midiSendProc

A pointer to the `MusicMIDISendProc` callback to use when sending MIDI data.

refCon

A reference constant value. The Movie Toolbox passes this reference constant to your callback each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This call is implemented only by music components for MIDI synthesizers.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicSetOfflineTimeTo

Advances the synthesizer clock when the synthesizer is not running in real time. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetOfflineTimeTo (
   MusicComponent mc,
   long newTimeStamp
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

newTimeStamp
The number of samples to synthesize.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The synthesizer may not be running in real time due to a call to `MusicStartOffline` (page 82). Setting the time generates audio output from the synthesizer.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicSetPart

Sets the MIDI channel and maximum polyphony for a specified part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPart (
   MusicComponent mc,
   long part,
   long midiChannel,
   long polyphony
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

Part whose MIDI channel and polyphony are to be set.

midiChannel

The MIDI channel to set the part to. For non-MIDI devices, set this parameter to 0.

polyphony

The maximum number of voices or polyphony for the part.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicSetPartAtomicInstrument

Initializes a part with an atomic instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartAtomicInstrument (
   MusicComponent mc,
   long part,
   AtomicInstrumentPtr aiP,
   long flags
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

The part to initialize with the atomic instrument to.

aiP

The atomic instrument.

flags

Constants (see below) that specify details of initializing a part with an atomic instrument. See these constants:

```
kGetAtomicInstNoExpandedSamples
kGetAtomicInstNoOriginalSamples
kGetAtomicInstNoSamples
kGetAtomicInstNoKnobList
kGetAtomicInstNoInstrumentInfo
kGetAtomicInstOriginalKnobList
kGetAtomicInstAllKnobs
```

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicSetPartController

Initializes the value of a specified controller on a specified part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartController (
   MusicComponent mc,
   long part,
   MusicController controllerNumber,
   long controllerValue
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

Part whose controller value you want to set.

controllerNumber

Controller number; see `Music Controllers`.

controllerValue

Value for controller.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicSetPartInstrumentNumber

Superseded by MusicSetPartInstrumentNumberInterruptSafe. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartInstrumentNumber (
   MusicComponent mc,
   long part,
   long instrumentNumber
);
```

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

MusicSetPartInstrumentNumberInterruptSafe

Initializes a part with a particular instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartInstrumentNumberInterruptSafe (
   MusicComponent mc,
   long part,
   long instrumentNumber
);
```

Parameters

mc
Music component instance identifier returned by NAGetRegisteredMusicDevice (page 91).

part
Part to be initialized.

instrumentNumber
Number of instrument to initialize part with. You can use MusicFindTone (page 53) to get an instrument number.

Return Value
See Error Codes. Returns noErr if there is no error.

Special Considerations

You can call this function at interrupt time.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicSetPartKnob

Sets a knob for a specified part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartKnob (
   MusicComponent mc,
   long part,
   long knobID,
   long knobValue
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part
The part number.

knobID
The index or ID of the knob to be set.

knobValue
The value to set the knob to.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicSetPartName

Changes the name of an instrument in a specified part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartName (
   MusicComponent mc,
   long part,
   StringPtr name
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part

Part to apply name to.

name

A pointer to the name to apply to part.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
You might want to change the name of a modified instrument before saving it. The instrument name string is used by selection dialog and configuration information boxes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

MusicSetPartSoundLocalization

Passes sound localization data to a specified synthesizer part. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicSetPartSoundLocalization (
   MusicComponent mc,
   long part,
   Handle data
);
```

Parameters

mc

Music component instance identifier.

part

The part to pass the data to.

data

The sound localization data.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Use the functions described in this section to get and modify the master tuning of the synthesizer, to play off line, and to allow the music component to perform tasks it must perform at foreground task time.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicStartOffline

Informs the QuickTime music synthesizer that the music will not be played through the speakers. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicStartOffline (
   MusicComponent mc,
   unsigned long *numChannels,
   UnsignedFixed *sampleRate,
   unsigned short *sampleSize,
   MusicOfflineDataUPP dataProc,
   long dataProcRefCon
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

numChannels
Number of channels in the music sample; 1 indicates monaural, 2 indicates stereo.

sampleRate
The number of samples per second.

sampleSize
The size of the music sample: 8-bit or 16-bit.

dataProc
A pointer to a `MusicOfflineDataProc` callback to handle the audio data.

dataProcRefCon
A reference constant to pass to the `MusicOfflineDataProc` callback. Use this parameter to point to a data structure containing any information your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Audio data will be sent to a function that will create a sound file to be played back later. You pass this function the requested values for the `numChannels`, `sampleRate`, and `sampleSize` parameters. When the function returns, those parameters contain the actual values used.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicStorePartInstrument

Puts whatever instrument is on the specified part into the synthesizer's instrument store. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicStorePartInstrument (
   MusicComponent mc,
   long part,
   long instrumentNumber
);
```

Parameters

mc
Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

part
Part containing the instrument to be stored.

instrumentNumber
Instrument number at which to store the part. The value must be between 1 and the synthesizer's modifiable instrument count, as defined by the `modifiableInstrumentCount` field of the synthesizer's `SynthesizerDescription` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function lets you store modified instruments.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

MusicTask

Allows a music component to perform tasks it must perform at foreground task time. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicTask (
   MusicComponent mc
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function must be called periodically. In the case of the QuickTime music synthesizer, instruments cannot be loaded from disk at interrupt time, so if the `NASetInstrumentNumberInterruptSafe` (page 104) function is called, the instrument is loaded during the next `MusicTask` call.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

MusicUseDeviceConnection

Tells a music component which hardware synthesizer to talk to. (Deprecated in Mac OS X v10.5.)

```
ComponentResult MusicUseDeviceConnection (
   MusicComponent mc,
   long id1,
   long id2
);
```

Parameters

mc

Music component instance identifier returned by `NAGetRegisteredMusicDevice` (page 91).

id1

The ID of the device returned in the id1 parameter of `MusicGetDeviceConnection` (page 58).

id2

The ID of the device returned in the id2 parameter of `MusicGetDeviceConnection` (page 58).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This call is implemented only for hardware synthesizers, such as PCI card devices.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NACopyrightDialog

Displays a copyright dialog box with information specific to a music device. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NACopyrightDialog (
   NoteAllocator na,
   PicHandle p,
   StringPtr author,
   StringPtr copyright,
   StringPtr other,
   StringPtr title,
   ModalFilterUPP filterProc,
   long refCon
);
```

Parameters

na
: You obtain the note allocator identifier by calling `OpenComponent`.

p
: A handle to a `Picture` structure containing the image resource for the dialog box.

author
: A pointer to a string containing author information.

copyright
: A pointer to a string containing copyright information.

other
: A pointer to a string containing any additional information.

title
: A pointer to a string containing title information.

filterProc
: Pointer to a `ModalFilterProc` callback.

refCon
: A reference constant value. The Movie Toolbox passes this reference constant to your `ModalFilterProc` each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

NADisposeNoteChannel

Deletes a specified note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NADisposeNoteChannel (
   NoteAllocator na,
   NoteChannel noteChannel
);
```

Parameters

na

You obtain the note allocator identifier by calling OpenComponent.

noteChannel

Note channel to be disposed. You obtain the note channel identifier from NANewNoteChannel (page 93) or NANewNoteChannelFromAtomicInstrument (page 93).

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
QuickTimeMusic.h

NAFindNoteChannelTone

Locates the instrument that best fits a requested tone description for a specific channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAFindNoteChannelTone (
   NoteAllocator na,
   NoteChannel noteChannel,
   ToneDescription *td,
   long *instrumentNumber
);
```

Parameters

na

You obtain the note allocator identifier by calling OpenComponent.

noteChannel

The note channel for which you want an instrument. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

td

A `ToneDescription` structure that describes the instrument fit.

instrumentNumber

On return, the number of the instrument that best fits the tone description.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAGetController

Retrieves the controller settings for a note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetController (
   NoteAllocator na,
   NoteChannel noteChannel,
   long controllerNumber,
   long *controllerValue
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel for which to get controller settings. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

controllerNumber

The controller for which to get settings; see `Music Controllers`.

controllerValue

On return, the value for the controller setting, typically 0 (0x00.00) to 32767 (0x7F.FF).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

NAGetIndNoteChannel

Returns the number of note channels handled by the specified note allocator instance. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetIndNoteChannel (
   NoteAllocator na,
   long index,
   NoteChannel *nc,
   long *seed
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

index

The index of the note channel. If 0, the result is still the number of note channels, but the `nc` parameter is not filled out.

nc

The note channel requested.

seed

A number that changes on successive calls if anything significant changes about a note channel; for example, if the note channel has been reallocated or released.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function can also return a requested note channel. To get a count of the note channels, pass 0 in the `index` parameter. To get a specific note channel, pass the index value returned by a previous call to `NAGetIndNoteChannel`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

NAGetKnob

Obtains the value of a knob for a given note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetKnob (
   NoteAllocator na,
   NoteChannel noteChannel,
   long knobNumber,
   long *knobValue
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

The note channel whose knob value you want to get. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

knobNumber

The index or ID of the knob whose value you want to get.

knobValue

On return, a pointer to the value of the knob.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAGetMIDIPorts

The MIDI input and output ports available to a note allocator. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetMIDIPorts (
   NoteAllocator na,
   QTMIDIPortListHandle *inputPorts,
   QTMIDIPortListHandle *outputPorts
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

inputPorts

On return, a handle giving the number of input ports (the first two bytes) followed by a list of `QTMIDIPort` structures.

outputPorts

On return, a handle giving the number of output ports (the first two bytes) followed by a list of `QTMIDIPort` structures.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This routine calls the QuickTime MIDI components to query them.

Special Considerations
`NAGetMIDIPorts` is the correct call for applications to make. They should not call `QTMIDIGetMIDIPorts`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAGetNoteChannelInfo

Returns the index of the music component for the allocated channel and its part number on that music component. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetNoteChannelInfo (
   NoteAllocator na,
   NoteChannel noteChannel,
   long *index,
   long *part
);
```

Parameters

na
You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel
Note channel to get information about. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

index
Music component index.

part
Music component part pointer.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The `NAGetNoteChannelInfo` function allows direct access to the music component allocated to the note channel by the note allocator. The index returned becomes invalid if music components are subsequently registered or unregistered.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

NAGetNoteRequest

Retrieves the NoteRequest structure that was passed to a note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetNoteRequest (
   NoteAllocator na,
   NoteChannel noteChannel,
   NoteRequest *nrOut
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

noteChannel

The note channel whose note request you want to get. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

nrOut

On return, the `NoteRequest` structure that was used when the specified note channel was allocated.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAGetRegisteredMusicDevice

Returns details about music components registered to the specified note allocator instance. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAGetRegisteredMusicDevice (
   NoteAllocator na,
   long index,
   OSType *synthType,
   Str31 name,
   SynthesizerConnections *connections,
   MusicComponent *mc
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

index

The index of the music component to get information about. To get a count of the registered music components, pass 0 in the `index` parameter. The return value is the count of components. To get information about one of the music components registered with the note allocator, pass the music component index in the `index` parameter. The index value can be 1 through the number of registered components returned by a previous call to `NAGetRegisteredMusicDevice`.

synthType

Synthesizer type.

name

Synthesizer name as a text string.

connections

A synthesizer connections for MIDI devices structure.

mc

Music component instance identifier.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

If you request information about a specific registered music component, this function returns the type of synthesizer the component supports in the `synthType` parameter, the name of the synthesizer in the name parameter, and the music component identifier in the `mc` parameter. For MIDI devices, it returns a pointer to a MIDI devices structure with information about the synthesizer connections.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

QTMusicToo

Declared In

`QuickTimeMusic.h`

NANewNoteChannel

Requests a new note channel with the qualities described in a NoteRequest structure. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NANewNoteChannel (
   NoteAllocator na,
   NoteRequest *noteRequest,
   NoteChannel *outChannel
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

noteRequest

A pointer to a `NoteRequest` structure.

outChannel

On return, a pointer to an identifier for a new note channel or `NIL` if the function fails to create a note channel.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function searches all available music components for the instrument that best matches the specifications in the `ToneDescription` structure that is contained within the `noteRequest` parameter. If an error occurs, the new note channel is initialized to `NIL`. The caller can request an instrument that is not currently allocated to a part. In that case, this function may return a value in `outChannel`, even though the request cannot initially be satisfied. The note channel may become valid at a later time, as other note channels are released or other music components are registered.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

qtmusic

qtmusic.win

QTMusicToo

Declared In

`QuickTimeMusic.h`

NANewNoteChannelFromAtomicInstrument

Requests a new note channel for an atomic instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NANewNoteChannelFromAtomicInstrument (
   NoteAllocator na,
   AtomicInstrumentPtr instrument,
   long flags,
   NoteChannel *outChannel
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

instrument

A pointer to the atomic instrument. This may be a dereferenced locked QT atom container.

flags

Flags (see below) that specify details of initializing a part with an atomic instrument. See these constants:

`kSetAtomicInstKeepOriginalInstrument`

`kSetAtomicInstShareAcrossParts`

`kSetAtomicInstCallerTosses`

`kSetAtomicInstDontPreprocess`

outChannel

On return, a pointer to an identifier for a new note channel or `NIL` if the function fails to create a note channel.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function takes a note allocator identifier in the `na` parameter and a pointer to the atomic instrument you are requesting a new channel for in the `instrument` parameter. Among other things, you can specify how to handle the expanded sample with the `flags` parameter. The function returns the note channel allocated for the instrument in the `outChannel` parameter, or `NIL` if an error occurs.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

qtmusic

qtmusic.win

Declared In

`QuickTimeMusic.h`

NAPickArrangement

Displays a dialog box to allow instrument selection. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAPickArrangement (
   NoteAllocator na,
   ModalFilterUPP filterProc,
   StringPtr prompt,
   long zero1,
   long zero2,
   Track t,
   StringPtr songName
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

filterProc

A Universal Procedure Pointer to a `ModalFilterProc` callback.

prompt

A pointer to a dialog box prompt string.

zero1

Must be 0.

zero2

Must be 0.

t

The arrangement movie track number.

songName

A pointer to the name of a song to display in the dialog box.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAPickEditInstrument

Presents a user interface for changing the instrument in a live note channel or modifying an atomic instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAPickEditInstrument (
   NoteAllocator na,
   ModalFilterUPP filterProc,
   StringPtr prompt,
   long refCon,
   NoteChannel nc,
   AtomicInstrument ai,
   long flags
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

filterProc

Pointer to a `ModalFilterProc` callback.

prompt

Dialog box prompt "New Instrument".

refCon

A reference constant value. The Movie Toolbox passes this reference constant to your `ModalFilterProc` callback each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

nc

The live note channel that appears in the dialog box. If you specify a note channel, set the `ai` parameter to 0. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

ai

The atomic instrument that appears in the dialog box. If you specify an atomic instrument, set the `nc` parameter to 0.

flags

Flags (see below) that limit the instruments presented. If the `kPickDontMix` flag is set, the dialog box does not display a mix of synthesizer part types. For example, if the current instrument is a drum, only available drums appear in the dialog box. The `kPickSameSynth` flag allows selections only within the current synthesizer. The `kPickUserInsts` flag allows user modifiable instruments to appear. If the `kPickEditAllowPick` flag is not set, no dialog box appears. See these constants:

```
kPickDontMix
kPickSameSynth
kPickUserInsts
kPickEditAllowPick
```

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

NAPickInstrument

Presents a user interface for picking an instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAPickInstrument (
   NoteAllocator na,
   ModalFilterUPP filterProc,
   StringPtr prompt,
   ToneDescription *sd,
   unsigned long flags,
   long refCon,
   long reserved1,
   long reserved2
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

filterProc

Pointer to a `ModalFilterProc` callback.

prompt

A pointer to the dialog box prompt "New Instrument".

sd

On entry, the tone description of the instrument that appears in the picker dialog box. On return, a tone description of the instrument the user selected.

flags

Flags (see below) that determine whether to display the picker dialog box and what instruments appear for selection. If the `kPickDontMix` flag is set, the dialog box does not display a mix of synthesizer part types. For example, if the current instrument is a drum, only available drums appear in the dialog box. The `kPickSameSynth` flag allows selections only within the current synthesizer. The `kPickUserInsts` flag allows user modifiable instruments to appear. The `kPickEditAllowPick` flag is used only with `NAPickEditInstrument` (page 95). See these constants:

```
kPickDontMix
kPickSameSynth
kPickUserInsts
```

refCon

A reference constant value. The Movie Toolbox passes this reference constant to your `ModalFilterProc` callback each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

reserved1

Must contain 0.

reserved2

Must contain 0.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
`QuickTimeMusic.h`

NAPlayNote

Plays a note with a specified pitch and velocity on the specified note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAPlayNote (
   NoteAllocator na,
   NoteChannel noteChannel,
   long pitch,
   long velocity
);
```

Parameters

na

You obtain the note allocator identifier from `OpenComponent`.

noteChannel

The note channel to play the note. You obtain the note channel identifier from the `NANewNoteChannel` (page 93) or the `NANewNoteChannelFromAtomicInstrument` (page 93) function.

pitch

The pitch at which to play the note. You can specify values as integer pitch values (0-127 where 60 is middle C) or fractional pitch values (256 (0x1.00) through 32767 (0x7F.FF)). If the pitch is a number from 0 to 127, then it is the MIDI pitch, where 60 is middle C. If the pitch is a positive number above 65535, then the value is a fixed-point pitch value. Thus, microtonal values can be specified. Negative values are not defined and should not be used.

velocity

The velocity with which the key is struck. Typically, this translates directly to volume, but on many synthesizers this also subtly alters the timbre of the tone. A value of 0 is silence; a value of 127 is maximum force.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win

QTMusicToo

Declared In
`QuickTimeMusic.h`

NAPrerollNoteChannel

Attempts to reallocate the note channel if it was invalid previously. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAPrerollNoteChannel (
   NoteAllocator na,
   NoteChannel noteChannel
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel to be re-allocated. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The `NAPrerollNoteChannel` function attempts to reallocate the note channel, if it was invalid previously. It could have been invalid if there were no available voices on any registered music components when the note channel was created.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

NARegisterMusicDevice

Registers a music component with the note allocator. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NARegisterMusicDevice (
   NoteAllocator na,
   OSType synthType,
   Str31 name,
   SynthesizerConnections *connections
);
```

Parameters

na

You obtain the note allocator identifier from `OpenComponent`.

synthType

Subtype of the music component.

name

The synthesizer name. This parameter provides a means of distinguishing multiple instances of the same type of device and is a string that can be displayed to the user. If no value is passed in the name parameter, the name defaults to the name of the music component type. The name appears in the instrument picker dialog box.

connections

A `SynthesizerConnections` structure that describes the hardware connections to a MIDI device.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAResetNoteChannel

Turns off all currently active notes on the note channel and resets all controllers to their default values. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAResetNoteChannel (
   NoteAllocator na,
   NoteChannel noteChannel
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

The note channel to reset. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function resets the specified note channel by turning "off" any note currently playing. All controllers are reset to their default state. The effects of the `NAResetNoteChannel` call are propagated down to the allocated part within the appropriate music component.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

NASaveMusicConfiguration

Saves the current list of registered devices to a file. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASaveMusicConfiguration (
   NoteAllocator na
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The `NASaveMusicConfiguration` function saves the current list of registered devices to a file. This file is read whenever a note allocator connection is opened, restoring the previously configured list of devices. The list is saved in the QuickTime Preferences file.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NASendMIDI

Sends a MIDI music packet to a synthesizer that contains a specific note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASendMIDI (
   NoteAllocator na,
   NoteChannel noteChannel,
   MusicMIDIPacket *mp
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

noteChannel

The function sends the packet to the synthesizer that contains this note channel. You obtain the note channel identifier from the `NANewNoteChannel` (page 93) or the `NANewNoteChannelFromAtomicInstrument` (page 93) function.

mp

The music packet to be sent.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function sends the MIDI music packet pointed to by the `mp` parameter to the synthesizer that contains the note channel identified by the `noteChannel` parameter. The `na` parameter specifies the note allocator instance to use.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

NASetAtomicInstrument

Initializes a synthesizer part with an atomic instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetAtomicInstrument (
   NoteAllocator na,
   NoteChannel noteChannel,
   AtomicInstrumentPtr instrument,
   long flags
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

The note channel to apply the atomic instrument to. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

instrument

A pointer to the atomic instrument. This can be a locked, dereferenced atomic instrument.

flags

Flags (see below) that detail how to initialize the part. See these constants:

```
kSetAtomicInstKeepOriginalInstrument
kSetAtomicInstShareAcrossParts
kSetAtomicInstCallerTosses
kSetAtomicInstDontPreprocess
```

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NASetController

Changes the controller setting on a note channel to a specified value. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetController (
   NoteAllocator na,
   NoteChannel noteChannel,
   long controllerNumber,
   long controllerValue
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel on which to change controller. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

controllerNumber

The controller to set; see `Music Controllers`.

controllerValue

Value for controller setting; typically 0 (0x00.00) to 32767 (0x7F.FF).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
`QuickTimeMusic.h`

NASetInstrumentNumber

Initializes initializes a synthesizer part with the specified instrument. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetInstrumentNumber (
   NoteAllocator na,
   NoteChannel noteChannel,
   long instrumentNumber
);
```

Parameters

na
You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel
Note channel to initialize with the instrument. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

instrumentNumber
Number of the instrument to initialize the part with. This number is unique to each synthesizer. General MIDI synthesizers all share the range 1-128 and 16365 to `kLastDrumKit`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NASetInstrumentNumberInterruptSafe

Initializes a synthesizer part with the specified instrument during interrupt time. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetInstrumentNumberInterruptSafe (
   NoteAllocator na,
   NoteChannel noteChannel,
   long instrumentNumber
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel to initialize with the instrument. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

instrumentNumber

Number of the instrument to initialize the part with.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

If the instrument is not already loaded when you call `NASetInstrumentNumberInterruptSafe`, you have to wait for the next call to `NATask` (page 109) for the instrument to become available.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

NASetKnob

Sets a note channel knob to a particular value. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetKnob (
   NoteAllocator na,
   NoteChannel noteChannel,
   long knobNumber,
   long knobValue
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel on which to set the knob value. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

knobNumber

Index or ID of the knob to be set.

knobValue
Value to set knob to.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

NASetNoteChannelBalance

Modifies the pan controller setting for a note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetNoteChannelBalance (
   NoteAllocator na,
   NoteChannel noteChannel,
   long balance
);
```

Parameters

na
You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel
The note channel to be balanced. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

balance
Specifies how to modify the pan controller setting. Valid values are from -128 to 128 for left to right balance.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NASetNoteChannelSoundLocalization

Passes sound localization data to a note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetNoteChannelSoundLocalization (
   NoteAllocator na,
   NoteChannel noteChannel,
   Handle data
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

The note channel to pass the data to. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

data

Sound localization data.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

NASetNoteChannelVolume

Sets the volume on the specified note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NASetNoteChannelVolume (
   NoteAllocator na,
   NoteChannel noteChannel,
   Fixed volume
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

noteChannel

The note channel to reset. You obtain the note channel identifier from the `NANewNoteChannel` (page 93) or the `NANewNoteChannelFromAtomicInstrument` (page 93) function.

volume

A fixed 16.16 number. `NASetNoteChannelVolume` sets the volume for the note channel, which is different from a `kControllerVolume` setting. Both volume settings allow fractional values of 0.0 to 1.0. Each value modifies the other. For example, a `kControllerVolume` value of 0.5 and a `NASetNoteChannelVolume` value of 0.5 result in a 0.25 volume level.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win

Declared In
`QuickTimeMusic.h`

NAStuffToneDescription

Initializes a tone description structure with the details of a General MIDI note channel. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAStuffToneDescription (
   NoteAllocator na,
   long gmNumber,
   ToneDescription *td
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

gmNumber

A General MIDI instrument number.

td

On return, an initialized tone description. The instrument name field will be filled in with the string name for the instrument.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
QuickTimeMusic.h

NATask

Called periodically to allow the note allocator to perform tasks in foreground task time. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NATask (
   NoteAllocator na
);
```

Parameters

na

You obtain the note allocator identifier from the Component Manager's `OpenComponent` function.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The `NATask` function calls each registered music component's `MusicTask` (page 83) function.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
QuickTimeMusic.h

NAUnregisterMusicDevice

Removes a previously registered music component from the note allocator. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAUnregisterMusicDevice (
   NoteAllocator na,
   long index
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

index

Synthesizer to unregister. The value is 1 through the registered music component count returned by `NAGetRegisteredMusicDevice` (page 91).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NAUnrollNoteChannel

Marks a note channel as available to be stolen. (Deprecated in Mac OS X v10.5.)

```
ComponentResult NAUnrollNoteChannel (
   NoteAllocator na,
   NoteChannel noteChannel
);
```

Parameters

na

You obtain the note allocator identifier by calling `OpenComponent`.

noteChannel

Note channel to be unrolled. You obtain the note channel identifier from `NANewNoteChannel` (page 93) or `NANewNoteChannelFromAtomicInstrument` (page 93).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

NewMusicMIDISendUPP

Allocates a Universal Procedure Pointer for the MusicMIDISendProc callback. (Deprecated in Mac OS X v10.5.)

```
MusicMIDISendUPP NewMusicMIDISendUPP (
   MusicMIDISendProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewMusicMIDISendProc`.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NewMusicOfflineDataUPP

Allocates a Universal Procedure Pointer for the MusicOfflineDataProc callback. (Deprecated in Mac OS X v10.5.)

```
MusicOfflineDataUPP NewMusicOfflineDataUPP (
   MusicOfflineDataProcPtr userRoutine
);
```

Parameters
`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewMusicOfflineDataProc`.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NewTuneCallBackUPP

Allocates a Universal Procedure Pointer for the TuneCallBackProc callback. (Deprecated in Mac OS X v10.5.)

```
TuneCallBackUPP NewTuneCallBackUPP (
   TuneCallBackProcPtr userRoutine
);
```

Parameters
`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewTuneCallBackProc`.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

NewTunePlayCallBackUPP

Allocates a Universal Procedure Pointer for the TunePlayCallBackProc callback. (Deprecated in Mac OS X v10.5.)

```
TunePlayCallBackUPP NewTunePlayCallBackUPP (
   TunePlayCallBackProcPtr userRoutine
);
```

Parameters

`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewTunePlayCallBackProc`.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

QTMIDIGetMIDIPorts

Returns two lists of MIDI ports supported by the specified MIDI component: a list of ports that can receive MIDI input and a list of ports that can send MIDI output. (Deprecated in Mac OS X v10.5.)

```
ComponentResult QTMIDIGetMIDIPorts (
   QTMIDIComponent ci,
   QTMIDIPortListHandle *inputPorts,
   QTMIDIPortListHandle *outputPorts
);
```

Parameters

ci

A MIDI component instance. Your software obtains this reference from `OpenComponent` or `OpenDefaultComponent`.

inputPorts

A list of the MIDI ports supported by the component that can receive MIDI input.

outputPorts

A list of the MIDI ports supported by the component that can send MIDI output.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The caller of this function must dispose of the `inputPorts` and `outputPorts` handles.

Special Considerations

`NAGetMIDIPorts` (page 89) is the correct call for applications to make. They should not call this function.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

QTMIDISendMIDI

Sends MIDI data to a MIDI port. (Deprecated in Mac OS X v10.5.)

```
ComponentResult QTMIDISendMIDI (
   QTMIDIComponent ci,
   long portIndex,
   MusicMIDIPacket *mp
);
```

Parameters

ci

A MIDI component instance. Your software obtains this reference from `OpenComponent` or `OpenDefaultComponent`.

portIndex

The index of the MIDI port to use for this operation.

mp

A pointer to the MIDI data packet to send.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function can be called at interrupt time. However, the same interrupt level is used whenever MIDI data is sent by the specified MIDI component.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

QTMIDIUseSendPort

Allocates a MIDI port for output or to release the port. (Deprecated in Mac OS X v10.5.)

```
ComponentResult QTMIDIUseSendPort (
   QTMIDIComponent ci,
   long portIndex,
   long inUse
);
```

Parameters

ci
A MIDI component instance. Your software obtains this reference from `OpenComponent` or `OpenDefaultComponent`.

portIndex
The index of the MIDI port for this operation.

inUse
Specifies whether to allocate the MIDI port for output (if the value is 1) or to release the port (if the value is 0).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneGetIndexedNoteChannel

Determines how many parts a tune is playing and which instrument is assigned to those parts. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetIndexedNoteChannel (
   TunePlayer tp,
   long i,
   NoteChannel *nc
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

i

Note channel index, or 0 to get the number of parts.

nc

A pointer to an allocated initialized note channel.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The tune player allocates note channels that best satisfy the requested instrument in the tune header. The application can use this call to determine which instrument was actually used for each note channel. This function takes the tune player in the `tp` parameter and returns the number of parts (1...n) allocated to the tune player. You can then pass the function a part index and it returns, in the `nc` parameter, the note channel allocated for that part.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

QTMusicToo

Declared In

`QuickTimeMusic.h`

TuneGetNoteAllocator

Returns the instance of the note allocator that the tune player is using. (Deprecated in Mac OS X v10.5.)

```
NoteAllocator TuneGetNoteAllocator (
   TunePlayer tp
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

Return Value
A note allocator or an error code. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneGetPartMix

Gets volume, balance, and mixing settings for a specified part of a tune. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetPartMix (
   TunePlayer tp,
   unsigned long partNumber,
   long *volumeOut,
   long *balanceOut,
   long *mixFlagsOut
);
```

Parameters

tp
A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

partNumber
The part number for this request.

volumeOut
Returns the volume for the part.

balanceOut
Returns the balance for the part.

mixFlagsOut
Returns flags (see below) that control part mixing. See these constants:
`kTuneMixMute`
`kTuneMixSolo`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneGetStatus

Returns an initialized structure describing the state of the tune player instance. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetStatus (
   TunePlayer tp,
   TuneStatus *status
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

status

A pointer to an initialized `TuneStatus` structure.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

qtmusic

qtmusic.win

Declared In

`QuickTimeMusic.h`

TuneGetTimeBase

Returns the time base of the tune player. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetTimeBase (
   TunePlayer tp,
   TimeBase *tb
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

tb

A pointer to a time base identifier, such as that returned by `NewTimeBase`. On return, the time base used to control the sequence timing.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The sequence can be controlled in several ways through its time base. The rate of playback can be changed, or the time base object can be slaved to a clock or time base different than real time.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneGetTimeScale

Returns the current time scale for a specified tune player instance. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetTimeScale (
   TunePlayer tp,
   TimeScale *scale
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

scale

A pointer to an initialized `TimeScale` variable that indicates the tune player's current time scale in units per second.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneGetVolume

Returns the volume associated with an entire tune sequence. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneGetVolume (
   TunePlayer tp
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

Return Value
The volume as a value from 0.0 to 1.0, or a negative result code. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneInstant

Plays a particular sequence of events active at a specified position. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneInstant (
   TunePlayer tp,
   unsigned long *tune,
   unsigned long tunePosition
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

tune

A pointer to tune sequence data.

tunePosition

The position within the tune sequence data in time units.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function plays the notes that are "on" at the point specified by the `tunePosition` parameter. The notes are started and then left playing on return. The notes can be silenced by calling `TuneStop` (page 128). This call is useful for enabling user "scrubbing" on a sequence.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

TunePreroll

Prepares to play a tune player sequence data by attempting to reserve note channels for each part in the sequence. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TunePreroll (
   TunePlayer tp
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneQueue

Places a sequence of music events into a queue to be played. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneQueue (
   TunePlayer tp,
   unsigned long *tune,
   Fixed tuneRate,
   unsigned long tuneStartPosition,
   unsigned long tuneStopPosition,
   unsigned long queueFlags,
   TuneCallBackUPP callBackProc,
   long refCon
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

tune

A pointer to an array of events, terminated by a marker event of subtype `kMarkerEventEnd`. See `QTMA Events`.

tuneRate

Speed at which to play the sequence. "Normal" speed is 0x00010000.

tuneStartPosition

Sequence starting time.

tuneStopPosition

Sequence stopping time. The `tuneStartPosition` and `tuneStopPosition` parameters specify, in time units numbered from 0 for the beginning of the sequence, which part of the queued sequence to play. To play all of it, pass 0 and 0xFFFFFFFF, respectively.

queueFlags

Flags (see below) with details about how to play the queued tunes. See these constants:

`kTuneStartNow`
`kTuneDontClipNotes`
`kTuneExcludeEdgeNotes`
`kTuneQuickStart`
`kTuneLoopUntil`
`kTuneStartNewMaster`

callBackProc

A pointer to a `TuneCallBackProc` callback.

refCon

A reference constant to be passed to your `TuneCallBackProc` callback. Use this parameter to point to a data structure containing any information your function needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneSetBalance

Modifies the pan controller setting for a tune player. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetBalance (
   TunePlayer tp,
   long balance
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

balance

A new pan controller setting. Valid values are from -128 to 128 for left-to-right balance.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneSetHeader

Prepares the tune player to accept subsequent music event sequences by defining one or more parts to be used by sequence Note events. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetHeader (
   TunePlayer tp,
   unsigned long *header
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

header

A pointer to a list of instruments that will be used in subsequent calls to the `TuneQueue` function. The list can include events with subtypes of `kGeneralEventNoteRequest`, `kGeneralEventPartKey`, `kGeneralEventAtomicInstrument`, `kGeneralEventMIDIChannel`, and `kGeneralEventUsedNotes`. It can also include atomic instruments. The list is terminated by a marker event of subtype `kMarkerEventEnd`. See `QTMA Events`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is the first QuickTime music architecture call to play a music sequence. The header parameter points to one or more initialized General events and atomic instruments. Only one call to this function is required. Each call to this function resets the tune player.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Related Sample Code
qtmusic
qtmusic.win
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneSetHeaderWithSize

Similar to TuneSetHeader but lets you specify the header length. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetHeaderWithSize (
   TunePlayer tp,
   unsigned long *header,
   unsigned long size
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

header

A pointer to a list of instruments that will be used in subsequent calls to the `TuneQueue` function. The list can include events with subtypes of `kGeneralEventNoteRequest`, `kGeneralEventPartKey`, `kGeneralEventAtomicInstrument`, `kGeneralEventMIDIChannel`, and `kGeneralEventUsedNotes`. It can also include atomic instruments. The list is terminated by a marker event of subtype `kMarkerEventEnd`. See `QTMA Events`.

size

The size of the header in bytes.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function resembles `TuneSetHeader` (page 122) in that it prepares the tune player to accept subsequent music event sequences by defining one or more parts to be used by sequence Note events. But unlike `TuneSetHeader`, it allows you to specify the header length in bytes. This prevents the call from parsing off the end if the music event sequence is missing an end marker.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneSetNoteChannels

Assigns note channels to a tune player. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetNoteChannels (
   TunePlayer tp,
   unsigned long count,
   NoteChannel *noteChannelList,
   TunePlayCallBackUPP playCallBackProc,
   long refCon
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

count

The number of note channels to assign.

noteChannelList

A pointer to the list of note channels to assign. The parts for the note channels you assign are numbered from 1 to the value of the `count` parameter.

playCallBackProc

A pointer to a `TunePlayCallBackProc` callback that is called for each event whose part number is greater than the value of the `count` parameter. Events whose part numbers are less than or equal to the value of the `count` parameter are passed to the note channel rather than the callback. This lets you to use the tune player as a general purpose timer/sequencer.

refCon

A reference constant to be passed to your callback. Use this parameter to point to a data structure containing any information your function needs.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

When you call this function, any note channels that were previously assigned to the tune player are no longer used and are disposed of.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

TuneSetPartMix

Sets volume, balance, and mixing settings for a specified part of a tune. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetPartMix (
   TunePlayer tp,
   unsigned long partNumber,
   long volume,
   long balance,
   long mixFlags
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

partNumber

The part number for this request.

volume

The volume for the part.

balance

The balance for the part.

mixFlags

Flags (see below) that control part mixing. See these constants:

`kTuneMixMute`

`kTuneMixSolo`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneSetPartTranspose

Modifies the pitch and volume of every note of a tune. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetPartTranspose (
   TunePlayer tp,
   unsigned long part,
   long transpose,
   long velocityShift
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

part

The part for which you want to change pitch and volume.

transpose
A value by which to modify the pitch of the note. The value is a small integer for semitones or an 8.8 fixed-point number for microtones.

velocityShift
A value to add to the `velocity` parameter passed to `NAPlayNote` (page 98).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneSetSofter

Adjusts the volume a tune is played at to the softer volume produced by QuickTime 2.1. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetSofter (
   TunePlayer tp,
   long softer
);
```

Parameters

tp
A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

softer
A value of 1 means play at the QuickTime 2.1 volume; a value of 0 means don't make the volume softer.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function adjusts the volume a tune is played at to the softer volume produced by QuickTime 2.1. Files imported with QuickTime 2.1 automatically play softer. Files imported with QuickTime 2.5 or later play at the new, louder volume.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneSetSoundLocalization

Passes sound localization data to a tune player. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetSoundLocalization (
   TunePlayer tp,
   Handle data
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

data

The sound localization data to be passed.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared In

`QuickTimeMusic.h`

TuneSetTimeScale

Sets the time scale used by the specified tune player instance. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetTimeScale (
   TunePlayer tp,
   TimeScale scale
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

scale

The time scale value to be used, in units per second.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function sets the time scale data used by the tune player's sequence data when interpreting time-based events.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code
QTMusicToo

Declared In
`QuickTimeMusic.h`

TuneSetVolume

Sets the volume for an entire sequence. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneSetVolume (
   TunePlayer tp,
   Fixed volume
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

volume

The volume to use for the sequence. The value is a fixed 16.16 number.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function sets the volume level of the active sequence to the value of the `volume` parameter, ranging from 0.0 to 1.0. Individual instruments within the sequence can maintain independent volume levels.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneStop

Stops a currently playing sequence. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneStop (
   TunePlayer tp,
   long stopFlags
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

stopFlags
Set to 0.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneTask

Lets a tune player to perform tasks it must perform at foreground task time. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneTask (
   TunePlayer tp
);
```

Parameters

tp
A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Call this function periodically to allow a tune player to perform certain operations it can performed only at foreground application task time. Specifically, the QuickTime music synthesizer cannot load instruments from disk at interrupt time. As a result, embedded program changes are not performed until this function is called.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`QuickTimeMusic.h`

TuneUnroll

Releases any note channel resources that may have been locked down by previous calls to TunePreroll for this tune player. (Deprecated in Mac OS X v10.5.)

```
ComponentResult TuneUnroll (
   TunePlayer tp
);
```

Parameters

tp

A tune player identifier, obtained from `OpenComponent` or `OpenDefaultComponent`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Related Sample Code

QTMusicToo

Declared In

`QuickTimeMusic.h`

Document Revision History

This table describes the changes to *QuickTime Music Architecture Reference*.

Date	Notes
2006-05-23	New document, based on previously published material, that describes the API for the QuickTime Music Architecture.

Document Revision History

Index

M

N

Q

S

T